Interactive Simulations of Biohybrid Systems

Sebastian von Mammen [1,*], **Daniel Wagner** [2], **Andreas Knote** [1], **and Umut Taskin** [2]

[1] *Human-Computer Interaction, University of Würzburg, Germany*
[2] *Organic Computing, University of Augsburg, Germany*

Correspondence*:
Sebastian von Mammen
University of Würzburg, Informatik 9, Am Hubland Süd, 97079 Würzburg, Germany,
sebastian.von.mammen@uni-wuerzburg.de

ABSTRACT

In this article we present approaches to interactive simulations of biohybrid systems. These simulations are comprised of two major computational components: (1) Agent-based developmental models that retrace organismal growth and unfolding of technical scaffoldings, and (2) interfaces to explore these models interactively. Simulations of biohybrid systems allow us to fast forward and experience their evolution over time based on our design decisions involving the choice, configuration and initial states of the deployed biological and robotic actors as well as their interplay with the environment. We briefly introduce the concept of swarm grammars, an agent-based extension of L-systems for retracing growth processes and structural artefacts. Next, we review an early augmented reality prototype for designing and projecting biohybrid system simulations into real space. In addition to models that retrace plant behaviours, we specify swarm grammar agents to braid structures in a self-organising manner. Based on this model, both robotic and plant-driven braiding processes can be experienced and explored in virtual worlds. We present an according user interface for use in virtual reality. As we present interactive models concerning rather diverse description levels, we only ensured their principal capacity for interaction but did not consider efficiency analyses beyond prototypic operation.We conclude this article with an outlook on future works on melding reality and virtuality to drive the design and deployment of biohybrid systems.

Keywords: Biohybrid Systems, Augmented Reality, Virtual Reality, User Interfaces, Biological Development, Generative Systems

Biohybrid systems, i.e. the cross-fertilisation of robotic entities and plants, take robotic control and interconnected technologies a significant step beyond the design, planning, manufacture and supply of complex products. Instead of pre-defined blueprints and manufacturing processes that fulfil certain target specifications, biohybrid systems consider, even make use of the variability of living organisms. By promoting and guiding the growth and development of plants, the characteristics exhibited throughout their life cycles become part of the system—from aesthetic greenery over load-bearing and energy-saving structural elements to the potential supply of nourishment. At the same time, biohybrid systems are feedback-controlled systems which means that (1) deviations of the individual plant, e.g. in terms of its health or developmental state, or (2) unexpected environmental trends, e.g. in terms of climatic conditions or regarding changes in the built environment, as well as (3) changes in the target specifications, can be compensated for. These traits of robustness, adaptivity and flexibility in combination with a potential longevity that may easily outlast a human lifetime, may very well render biohybrid systems a key technology in shaping the evolution of man-kind. However, comprehensive basic research has to be conducted in order to arrive at state mature enough to deploy and benefit from a biohybrid system outside of laboratory conditions.

While it may seem obvious that the primary concerns of an according research agenda are the properties and interactions of plants and robots, the resulting systems pose an intriguing challenge also in terms of user interaction. Their expectable non-linearity as well as their strong dependency of concrete environmental location and condition demand for an according comprehensive, flexible and location-dependent planning process: The designer should be empowered to travel to a prospective deployment site for a biohybrid system, investigate its various possible configurations in the given context and probe its potential impact over time. This capability can only be realised based on several technological requirements. It implies that given a specific parameter set, we need plausible predictions about the development of the system. Changes to these parameters need to be considered as well. In addition, the corresponding, dynamic simulation has to leave a small computational footprint so that the user can rely light-weight mobile computing devices to evaluate designs at the very locations where the biohybrid systems should be deployed. Furthermore, these simulations have to run at realtime speed so that various impact factors that the designer foresees can be considered in the scope of one or many simulations. Serving the simulated development of a biohybrid system in-situ not only challenges the systems' engineers in terms of computational efficiency—the in-situ projection also needs to be supported by an accessible user interface which considers the intricacies of biohybrid systems as well as the complexities of their physical environment.

In this article, we present our ongoing efforts towards according technologies at the intersection of biohybrid systems and their human users. Our goal is to simulate biohybrid systems in realtime and to make these simulations interactive. Prototyping, planning, and deployment of biohybrid system configurations represent the immediate use cases for the corresponding realtime interactive simulations. Accordingly, our approach considers realtime-capable simulation models of plant growth and dynamics as well as robotic interactions. Generative models such as L-Systems and generic agent-based modelling approaches paved the way for the models we devised for interactive biohybrid simulations. We briefly survey these preceding works in Section 1. Next, we introduce our interactive modelling approach for biohybrid systems in Section 2. More specifically, we adjust a swarm grammar representation to incorporate various developmental behaviours of plants such as lignification, phototropism and shade avoidance. We also utilise the agent-based swarm grammar approach to develop futuristic models of robotic units braiding scaffolding structures as currently worked on in the biohybrids research community. In Section 3, we present an augmented reality (AR) prototype for the design of biohybrid systems. The specific challenges introduced by the augmented reality setting, such as remodelling real-world lighting conditions or limited input capabilities are overcome in a virtual reality (VR) prototype presented in Section 4. Another advantage of VR is that the development and effect of a biohybrid system can be experienced in the context of arbitrary (virtual) environments, no matter how remote or futuristic they may be. For now, it also allows us to focus on the design of concrete user interfaces for selecting and configuring biohybrid components and to navigate through the simulation process. We conclude this article with a summary and an outlook on future work in this field.

1 GENERATIVE MODELS

At the core of the virtual or augmented, projected biohybrid system prototypes that we present in this article lie various generative models that drive the development and growth of robotic and plant-based structures. In this section, we summarise preceding works in the field of generative modelling. First, we briefly explain the general idea of procedural generation of content. Often, it is used in the context of creating computer graphics assets, for instance for three-dimensional terrains or detail-heavy textures. Next, we introduce L-systems, a generative modelling approach that translates basic biological proliferation into a formal representation which, in turn, can be geometrically interpreted and visualised (Prusinkiewicz and Hanan (2013)). L-systems define the state-of-the-art in generating three-dimensional assets of plants but are also widely applied in other contexts—from breeding novel hardware designs (Tyrrell and Trefzer (2015)) to the encoding of artificial neural networks (de Campos et al. (2015)). L-systems have been extended in various ways to support dependencies to the environment or specific behaviours in development, such as gravitropism or crawling. Swarm grammars represent the most open and flexible extension of L-systems as the plants' tips as well as the grown stem segments, leaves, etc. are considered agents that can react to their environment in arbitrary ways, also in realtime. Therefore, we chose swarm grammars as the basic representation for our interactive biohybrid experiments.

1.1 Procedural Content Generation

Interactive systems such as the ones that we present in this article are—from a perspective of technology—rather close to video and computer games. They have to calculate and render models at high speeds to ensure that there are no lags for the user's camera view(s). They also have to provide means for interaction and provide adequate responses both regarding the behaviours of the simulated system and its visualisation. Overall, the requirements for procedural content generation (PCG) approaches are very similar in games and in our application scenario. Shaker et al. (2014) detail PCG approaches that are frequently used in the context of computer games. They understand PCG as algorithmically creating contents, whereas user input only played a minor role, if involved at all. In particular, one distinguishes between utilising PCG to generate contents before a game is played or a simulation is run (online vs. offline). The PCG content is considered necessary if it plays an instrumental role in the interactive scenario. Otherwise, if it is only meant as eye-candy, it is optional. Depending on the information that is fed into the PCG machinery, the approach can be classified as either driven by random seeds or by parameter vectors that may determine one or the other parameter range or provide constant values. The way this data informs the PCG algorithm(s) may be deterministic, i.e. it reliably produces identical results at each run, or stochastic, and vary in its output accordingly. Furthermore, a PCG approach may be constructive which means that compliance with a certain goal or satisfaction of a set of given constraints is ensured while an artefact is created. The alternative would be to generate an artefact first and test whether it fulfills the required criteria afterwards, which is referred to as generate-and-test.

1.2 Functions, Reactions, Behaviours

Depending on the overarching goals, different approaches lend themselves better for generating contents than others. For instance, there are several methods for generating artificial landscape terrains—from smooth to rugged, even to sharp surfaces. Midpoint Displacement or Diamond-Square, for instance, are simple equations that recursively divide line segments to determine values on a height map dependent on neighbouring points (Rankin (2015)). A considerable improvement can further be achieved, when considering external forces such as erosion (Cristea and Liarokapis (2015)).

While physicality plays an enormous role, complex structures in nature often emerge from organismal behaviours. These may be simple, repetitive, reactive such as the habitual secretion of calcium deposits which results in the formation of molluscan shells or skeletons of the corals (Thompson (2008)). As soon as cellular proliferation and differentiation is considered, complex branching structures can emerge. L-systems are a computational representation that effectively abstract the complexity of organismal growth, yet are capable to retrace the development of complex forms (Lindenmayer (1971); Prusinkiewicz and Lindenmayer (1996)).

1.3 L-Systems

In L-systems, a symbol or character represents a biological cell at a specific state or of a specific type. Production rules, similar to those associated with formal grammars, determine which other state/type a cell will transition into in the next iteration. As a cell may also reproduce, a single cell may yield two or more new cells. Considering that transitions may not be fully deterministic, probabilities could be associated with such production rules. Some transitions might also be triggered by a cell's context. According context-sensitive rules require two or more cells to be present, possibly in a specific state, to trigger a transition, etc. There is a large number of variations of production rules and their respective impact on the emergent processes and artefacts. No matter which specific L-system one implements, they all have in common that based on an initial axiom and a set of production rules a string is iteratively generated by applying all fitting rules in parallel at each step of the algorithm.

At each iteration, the generated string can be interpreted graphically: The so-called "turtle interpretation" algorithm steps through the symbols of the string and considers them instructions for a turtle to turn and walk by a certain degree or distance. Figure 1 shows the first three steps of four different L-systems illustrated by means of the turtle interpretation. The production rules of L-systems substitute any symbols in accordance with the respective rules. For instance, an initial "A" might have been replaced by "AB" at the next step. For the turtle interpretation, the set of symbols may, for instance, include F for "forward", $-$ for "turn left", $+$ for "turn right", [for "remember position" and] for "resume last position". In Figure 1(a), a rather simple rule set repeatedly replaces the initial symbol, or axiom, A with FA, whereas A does not carry a graphical meaning. Next, in Figure 1(b), the orientation of the turtle is instructed by introducing hyphens. The L-system in Figure 1(c) is simply a bit more involved than (b), whereas in (d) the bracketing concept has been introduced, which results in according branching structures.

0L, 1L, and 2L-systems are the basic classes of L-systems discussed in light of Chomsky's hierarchy of formal languages. 0L-systems are context-free and do not consider interdependencies between individual cells, whereas the other classes offer production rules that consider the substitution of individuals cells in the context of one, respectively two neighbouring cells. As an example, the simplest, context-free l-system has rules $p \xrightarrow{\theta} s$, whereas $p \in \Omega$ is a symbol of an alphabet Ω and $s \in \Omega*$ represents a word over Ω or an empty symbol. With probability θ, p is substituted by s. There is a multitude of extensions to L-systems, including parameterised L-systems which introduce scalars into the otherwise symbolic rules. These values can, for instance, be used to encode continuous changes of organismal development. Parameterisation of the l-system rules also allows to introduce constraints that link developmental processes or let them interact with the environment.

Graph-based representations are rather expressive and in the 1970s, according extensions to l-systems were already presented (Culik and Lindenmayer (1976)). These efforts were resumed in the early 2000s to devise relational graph grammars (RGGs) a rather flexible implementation of the l-system idea (Kniemeyer et al. (2004)). In RGGs, parametric l-systems are extended with object-oriented, rule-based, procedural features, which allows to retrace various forms of l-systems, generating arbitrary cellular topologies, and even modelling other, rather process-oriented representations such as cellular automata or artificial chemistries. The integration of aspects of development and of interaction supports modelling organisms such as plants considering both their structure and their function (Kniemeyer et al. (2006, 2008)). L-Systems have, of course, already been used in the context of realtime interactive systems as well. In one particular

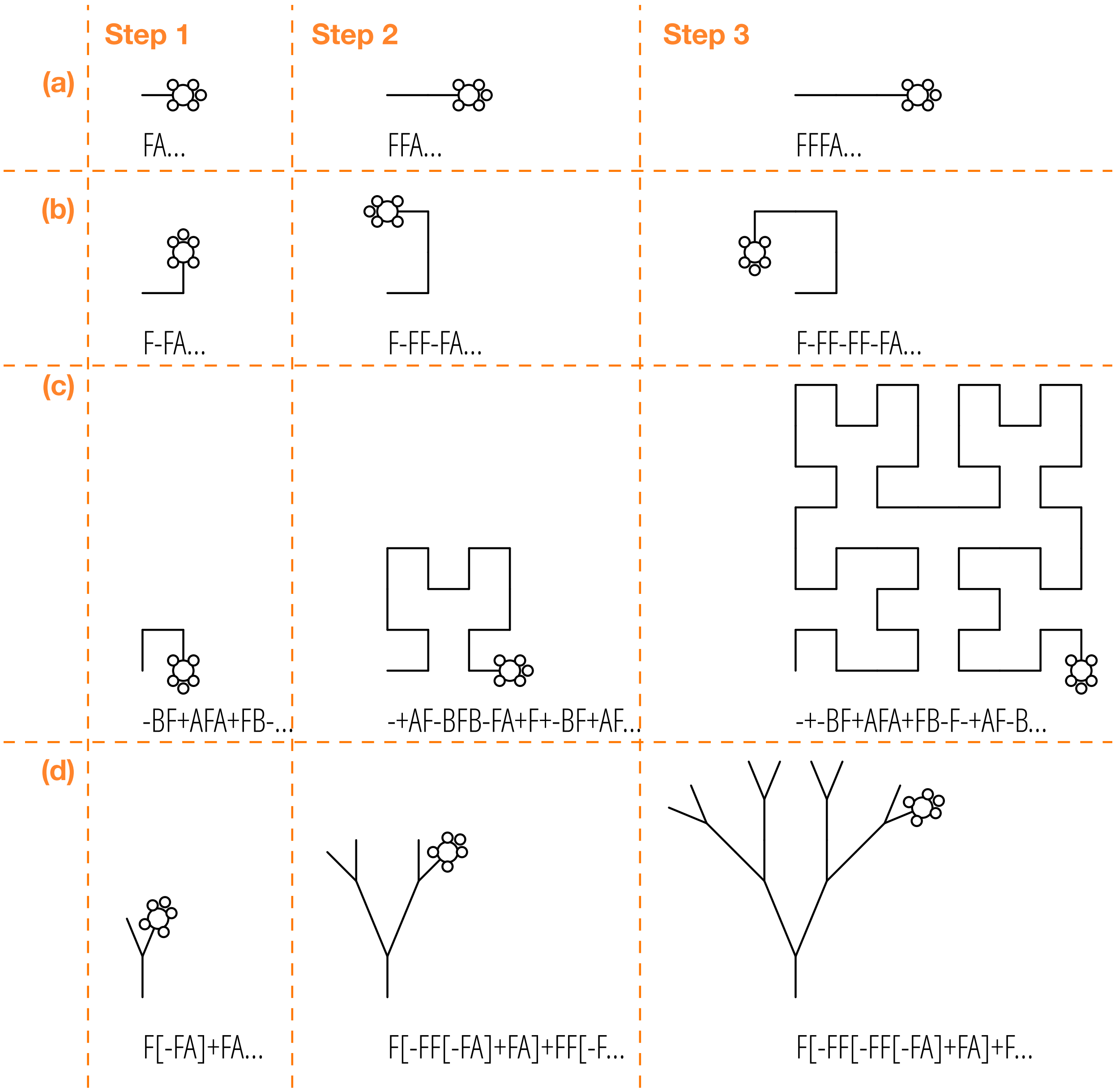


Figure 1. The first three production steps of four different L-systems (a) to (d) are illustrated by means of the turtle interpretation.

instance, an efficient implementation allows the user to generate strings and interpret them visually fast enough as to explore the model space and adjust the concrete instances' parameters for the growth within interactively selected regions of interest (Onishi et al. (2003)). This idea was resumed by Hamon et al. (2012), who made it possible not only to let the L-System grow based on contextual cues such as collisions but also to change the L-System formalism interactively, on-the-fly.

1.4 Agent-based Approaches

The turtle interpretation of l-systems as illustrated in Figure 1 simulates an agent (the turtle) leaving a trail, thereby creating an artefact. Agents receive sensory information about their environment, process the information and choose actions in accordance with their agenda. Such an agent-based perspective could, of course, drive the actual construction algorithm. According approaches have been proposed, for instance by Shaker et al. (2014). The authors demonstrate the concept in the context of a digger agent that leaves corridor trails with chambers at random points in a subsurface setting. They distinguish between an uninformed, “blind” agent and one that is more aware of the built environment only places new chambers that do not overlap with previously existing ones. Figure 2 shows this representative constructive agent-based example. In our implementation, both agent types had a chance of 10% of changing their direction and of 5% of creating a chamber at each step. The chamber dimensions were randomly chosen between 2 to 5 times of a single corridor cell.

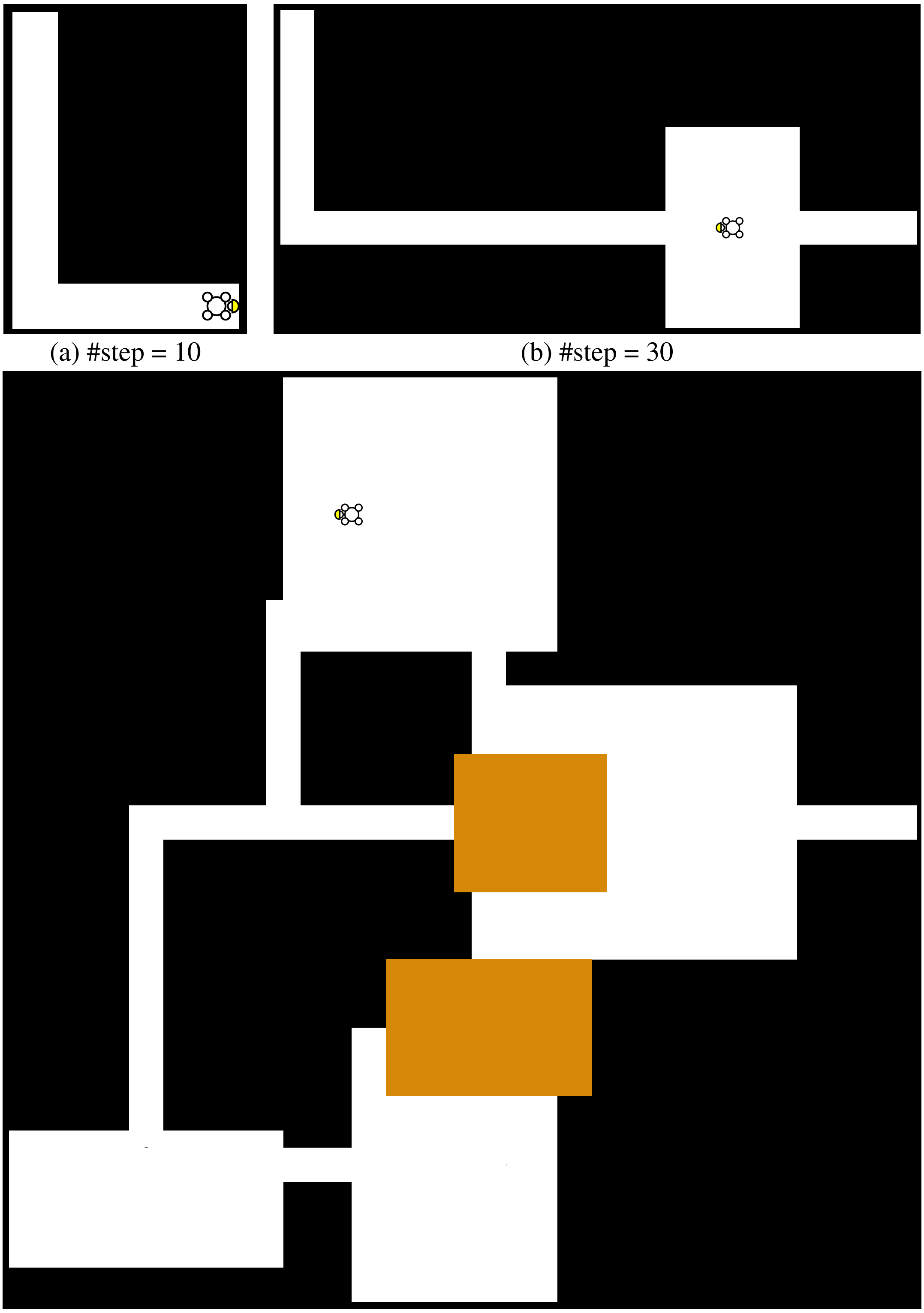
2 SWARM GRAMMARS

In l-systems and other grammatical developmental representations, neighbourhood topologies and neighbourhood constraints (who informs whom and how?) are mostly embedded in production rule sets. This is different for the agent-based approach, which also explains the need for awareness about the built environment to achieve coordinated constructions (see Figure 2). Overall, the quality of agent-generated artefacts greatly depends on the ingenuity and complexity of their behavioural programme and on the simulated environment.

An important advantage over grammatical representations such as l-systems is the simplicity of extending agent-based systems. Sensory information, behavioural logic or the repertoire of actions can be easily and directly changed. Dependencies to other agents or the environment can be designed relative to the agent itself and the topology among interaction partners can evolve arbitrarily based on a modelled, possibly dynamic environment and arbitrary preceding multi-modal interactions. The inherent flexibility of agents (due to threefold design of sensing/processing/acting) facilitates the resulting system to be interactive not only with respect to a modelled environment but also to user input that is provided on-the-fly.

Swarm grammars (SGs) bring together the agent-based, interactive and the reproductive, generative perspectives (von Mammen and Jacob (2009); von Mammen and Edenhofer (2014)). In the 1980s, Reynolds published on the simulation of flocks of virtual birds, or boids (Reynolds (1987)). Each boid is typically represented as a small, stretched tetrahedron or cone to indicate its current orientation. It perceives its neighbours within a limited field of view (often a segment of a sphere), and it accelerates based on its neighbours’ relative positions and velocities. A boid’s tendency towards the neighbours’ geometric centre, away from too close individuals and alignment of their velocities yields complex flock formations. Boids are very simple, so-called reactive agents that interact merely spatially. Due to their simplicity, they lend themselves well for a primary agent model to be extended by the l-system concept of generative production. As a result, SGs augment boids to leave trails in space and to differentiate and proliferate as instructed by a set of production rules.

Formally speaking, a swarm grammar $SG = \{SL, \Delta\}$ consists of a system SL that is comprised of an axiom α and a set of production rules P, and a set of agent types or agent specifications Δ. Each specification $\delta \in \Delta$ may determine an arbitrary set of agent features and their respective values. These features may, for instance, include specifics of the agents’ visual or spatial representation, relate to their states or behaviours. The rewrite system SL implements a probabilistic l-system as introduced in Section 1.3, whereas each symbol p of the production rules refers to the alphabet of agent specifications and all actual agent instances in the simulation are configured in accordance with their types. Different from the turtle interpretation of l-systems, tracing the movement of agents leads to structures and their reproduction yields branching. Figure 3 shows three basic swarm grammar implementations, relying on two agent types A and B. Both of them fly upwards (stepwise positional increment of 0.1 along the y-axis). In addition, B deviates at a 10% chance along the x- and z-axes at each step, with a random increment of maximal 0.5 in

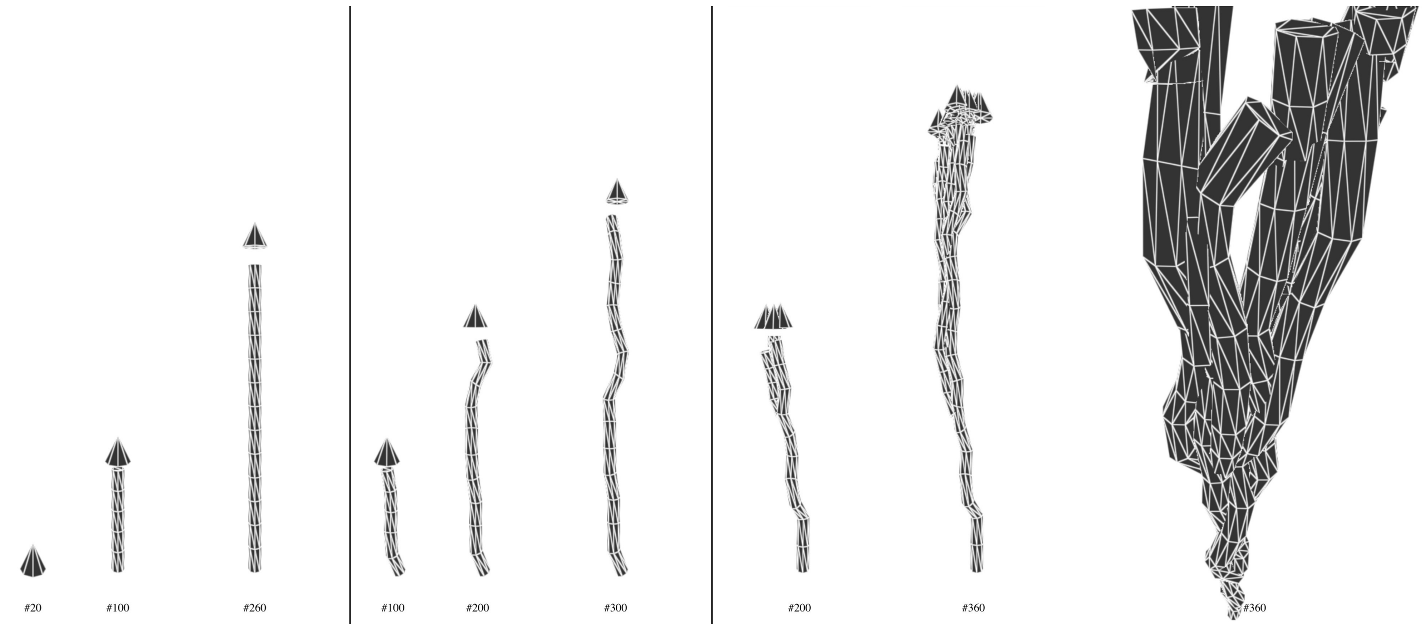


(a) #step = 10 (b) #step = 30

(c) #step = 100

Figure 2. Illustration of an agent-based constructive approach to procedural content generation. (a) When moving, the agent leaves corridor trails. (b) It changes its direction and creates chambers at randomly. (c) An uninformed agent creates overlapping chambers (in orange).

213 each direction. SL_1 only deploys agent specification A, SL_2 and SL_3 only specification B, and SL_3 also
214 lets its agents reproduce with a probability of 1% at each step.



$SL_1 = \{\alpha = A, P = \{\}\}$ | $SL_2 = \{\alpha = B, P = \{\}\}$ | $SL_3 = \{\alpha = B, P = \{B \xrightarrow{0.01} BB\}\}$

Figure 3. Three increasingly complex swarm grammars. The counters beneath the screenshots indicate the progression of the respective simulation. The rewrite systems SL express which agent configurations were deployed an whether and how they reproduced. The right-hand side image shows a close-up of the branched structure from SL_3.

To this date, most swarm grammar implementations incorporate the flocking model by Reynolds (1987), where simple local reactive acceleration rules of spatially represented agents drive the flight formation of agent collectives. Hence, next to attributes of the agents' display, e.g. their shape, scale and colour, the agent specifications δ also consider the parameterisation of the agents' fields of view and the coefficients that determine their accelerations with respect to their perceived neighbourhoods. In particular, these coefficients weigh several different acceleration "urges". These include one that drives an agent to the geometric centre of its peers (cohesion), one that adjusts its orientation and speed towards the average velocity of its peers (alignment), one that avoids peers that are too close (separation) as well as some stochasticity. The field of view that determines the agents' neighbourhood perception is typically realised by a viewing angle and by testing proximity (within a maximal perception distance, potentially triggering uneasy closeness).

Over the years, swarm grammars have evolved in different directions, some implementations featuring agents that individually carry the production rules along in order to rewrite them based on local needs or store/retrieve them alongside the agent's other data (von Mammen and Edenhofer (2014)). This modeling decision begs the question to identify the unique features of swarm grammars in contrast to general multi-agent systems (MAS), see, for instance Wooldridge (2009). Clearly, swarm grammars represent a subset of MAS. They can be reduced to MAS with state-changing interactions, type-changing differentiation and reproduction. Typically swarm grammars implement spatial interaction and yield structural artefacts. In the context of interactive simulations for planning, configuration, adjustment and exploration of biohybrid systems, swarm grammars pose an apt modeling and simulation approach due to their flexibility in terms of agent specifications, their generative expressiveness, and their capacity to (a) interact with complex virtual environments and (b) the user/operator in realtime.

2.1 An Interactive Growth Model

In the context of biohybrid systems, robots influence the growth and movement of plants by exploiting their reactive behaviours and dynamic, environment-dependent states. Precise simulation of the multitude of interaction possibilities and the resulting reactions by plants is not feasible, yet. However, one can model plants and their dynamics at an abstract level which is feasible to calculate at interactive speeds and which yields plausible outcomes of the plants' evolution. By means of swarm grammars, we can specify arbitrary agent properties, behaviours and production rules to drive a computational developmental model. Those processes that are observed and described at the level of a plant individual represent an adequate level of abstraction for realising interactive biohybrid system simulations. As an optimisation step, groups of individuals might be subsumed and be calculated as single meta-agents (von Mammen and Steghöfer (2014)) but individual plants are the basic unit of abstraction as their influence on the biohybrid system matters. Therefore, to let interactive swarm grammars retrace biological growth and dynamics more closely, we started incorporating various behavioural processes exhibited by different plants to different degrees. Among the most common behaviours are the growth of a plant, its movement, orientation towards light, avoidance of shadow, bending, and lignification (for a general introduction, see for instance Stern et al. (2003)). In the following paragraphs, we shed light on our implementations of these behaviours and the underlying, abstract models.

2.1.1 Articulated Plant Body

In the original swarm grammar model, there was a clear distinction between the static built artefact and the interacting, building agents (von Mammen (2006)). Later, these components were unified and arbitrary living agents or inanimate building blocks were placed by the simulated agents based on local interaction rules rather than grammatical production rules (von Mammen and Jacob (2009)). In order to retrace the dynamics of plant physiology, we decided to follow the original approach, keep the tip of our abstract plant model separate from the stem's segments and assign very clear capabilities to these primary and secondary data objects. In order to support the dynamics arising from interdependencies between the stem's segments, we introduced a hierarchical data structure to traverse the segments in both directions, also considering branches. This traversal is required to retrace the transport of water, sugar and other nutrients but also to provide a physical, so-called articulated body structure. Figure 4 shows a swarm grammar with rewrite system $SL_4 = \{\alpha = C, P = \{C \xrightarrow{0.01} CC\}\}$ after 660 simulation steps, whereas C extends agent B from SL_2 and SL_3 by means of a separation urge that accelerates away from peers that are closer than 10.0 units. The resulting spread of the branches allows one to retrace the hierarchical data structure annotated in the figure.

2.1.2 Iterative Growth

In our abstract plant model, the tip determines the direction of growth by moving upwards (gravitropism) and in accordance with the lighting situation (phototropism and shade avoidance). Growth is primarily realised by repeatedly adding segments to the plant's body that are registered as children in the hierarchy. The conceptual translation of biological growth to this additive process follows the original swarm grammar model which is shown in Figure 3. Secondarily, the segments grow in diameter, increasing the transport throughput, which is needed to supply new growth at the tip(s) of the plant. Water is transported up the Xylem vessels to the leaves, and together with sugar produced in the leaves travels back to the roots through the Phloem cell system, see for instance Fiscus (1975). In our model, these flows are abstractly captured as the exchange of information between the segments and the resulting expansion of the plant's body is reflected in the stem's diameter but also in the throughput. We assigned an according variable bandwidth b_i

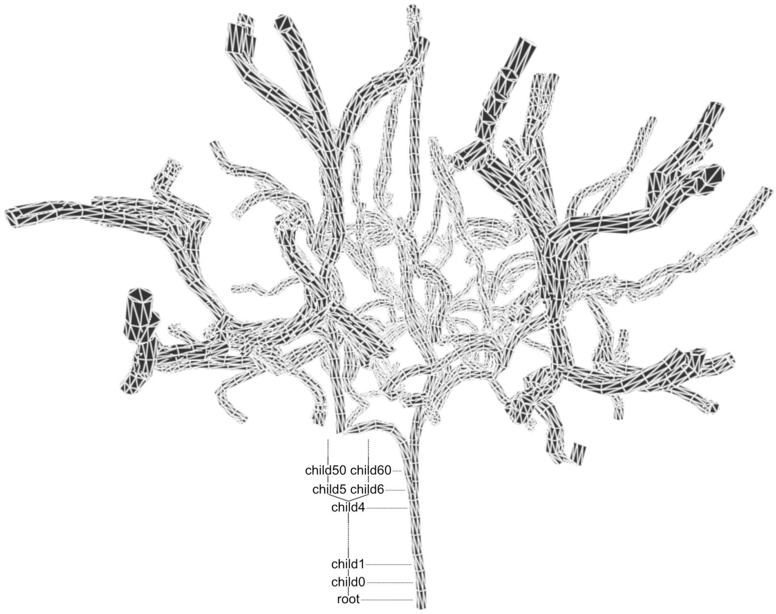


Figure 4. A tree structure that unfolds after 660 steps from a swarm grammar with rewrite system $SL_4 = \{\alpha = C, P = \{C \xrightarrow{0.01} CC\}\}$ and C implementing upwards flight, random movement across the xz-plane, probabilistic branching and separation. The artefact is captured as an articulated body by means of a hierarchical data structure starting at the root segment. The labels from the root up denote the respective segments' (referenced by the dotted horizontal lines). The perspective view slightly distorts the appearance of the uniformly scaled segments.

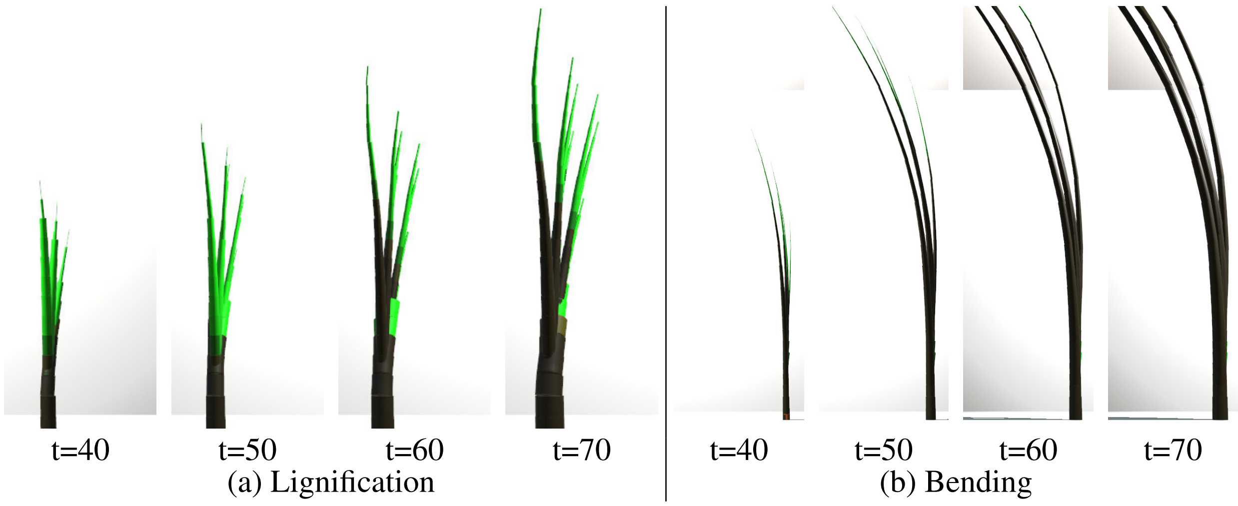


Figure 5. Screenshots of two swarm grammars specifically illustrating lignification and bending at simulation time step t.

to each segment i. In order to enable growth at the tip(s) of the plant, the concrete demand of supply is communicated downstream and the segments are expanded by an increment $\varDelta b$ recursively from the root upwards.

2.1.3 Lignification and Bending

In order to channel the increasing flows, the plant also needs to gain more structural integrity which is realised by the process of lignification. It means that the stem becomes more rigid and woody to gain more stability based on the deposition of lignin. We modelled this process by introducing an according state variable, stability s_i, for all plant segments i. Said hierarchical links between the segments make it possible to simulate the dynamics of the stem. Plants bend due to the weight of the stem, branches, flowers and fruits. In addition, environmental interactions, e.g. exposure to wind or collisions with other plants or objects, as well as the plant changing its direction of growth, may all contribute to bending the plant stem. In our model, we only consider the plant's own weight and the resulting forces. Other forces would need to be applied analogously. In order to achieve plausible bending of the plant in realtime, we consider an individual segment's stability s_i, and the segments bandwidth b_i, the number of total children of the segment n_i and the position of the tip of the branch, p_i^{Tip}. First, using the projection operation onto the XY plane $\mathbb{P}_{\text{XY}}$ and the unit vector e_Z in Z direction, a bending target direction $\mathbf{d}_i^t$ is calculated using Equation (1). Then, incorporating the stiffness and the integration time step, we compute the new orientation, a quaternion $\mathbf{R}_i'$ of the segment as the linear quaternion interpolation lerp between its current orientation and the influence by all of its children as summarised in Equation (2), whereas the function $\mathrm{rotation}$ yields a quaternion oriented towards a given vector. In Figure 5(a) the process of lignification is depicted by means of a branching swarm grammar. A simple colouring scheme is directly mapped to the hierarchy to illustrate the age and the degree of lignification of the respective segments. In Figure 5(b) the growth target of a swarm grammar is slightly shifted to the left. In this way, the plant bends based on its own weight.

$$\mathbf{d}_i^{\text{Tip}} = n_i b_i (1 - s_i)\, \mathbb{P}_{\text{XY}}(\mathbf{p}_i^{\text{Tip}} - \mathbf{p}_i) + 20\mathbf{e}_Z \tag{1}$$

$$\mathbf{R}_i' = \mathrm{lerp}(\mathbf{R}_i, \mathrm{rotation}(\mathbf{d}_i^{\text{Tip}}), s_i \Delta t) \tag{2}$$

2.1.4 Phototropism and Shadow Avoidance

According to the basic swarm grammar implementation, we expressed branching processes as production rules. Currently, exceeding a given nutritional value triggers the respective rules. Other conditions, for instance relating to the achieved form or considering pruning activities by a gardener, may trigger productions just the same. Phototropism, i.e. the urge to grow towards light, is realised as follows. A set of light sources is iterated and, if the respective light is activated, a raycast, i.e. a projected line between the two objects, reveals whether the light shines on a given segment or not: The raycast may not collide with other objects and the angle between the light source and the segment may not exceed the angle of radiation. In this case, the distance vector to the light source $\mathbf{d_i^l}$ down-scaled by some constant $c \in [0, 1]$, the stability factor s_i, and the segment's current position $\mathbf{p_i}$ determine the segment's re-orientation in accordance with Eqn. 3. In this way, the branch's growth, supported by its stability, is directed towards the light source. Gravitropic response is incorporated implicitly here, instead of an additional term that is eventually blended in. Figure 6 shows the visual artefact resulting from the tandem of lignification and phototropic growth. The raycast may also reveal that the segment is not lit by a given light source—similar to determining an object's shading based on shadow volumes, elaborated for instance by Wyman et al. (2016). In this case, the plant's gravitropic response is overwritten by a deflection that reduces upwards growth by 75%. The result can be seen in Figure 7: At $t = 0$, the plant picks up a ray from the top-left light source. A few steps later, the lights are toggled, and the sideways growth gets reaffirmed. Once the shadow yielding plate is overcome, at around $t = 100$, the gravitropism and phototropism boost the development of the different branches of the plant.

$$\mathbf{R}_i = \mathrm{lerp}(\mathbf{R}_i, \mathrm{rotation}(\mathbf{p}_i + c\mathbf{d}_i^l), s_i \Delta t) \tag{3}$$

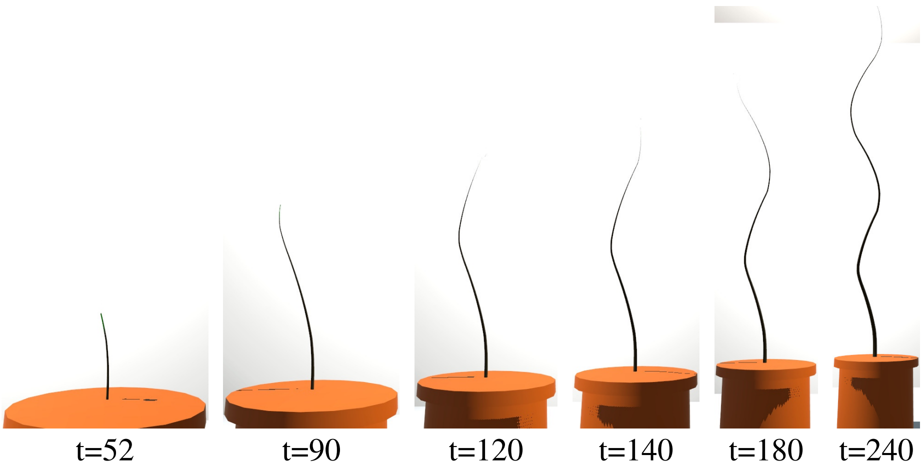


Figure 6. Screenshots of a swarm grammar specifically illustrating phototropic growth over time. Two light sources at the top-left and top-right are alternately activated to guide the movement of the tip and thereby influence the shape of the stem.

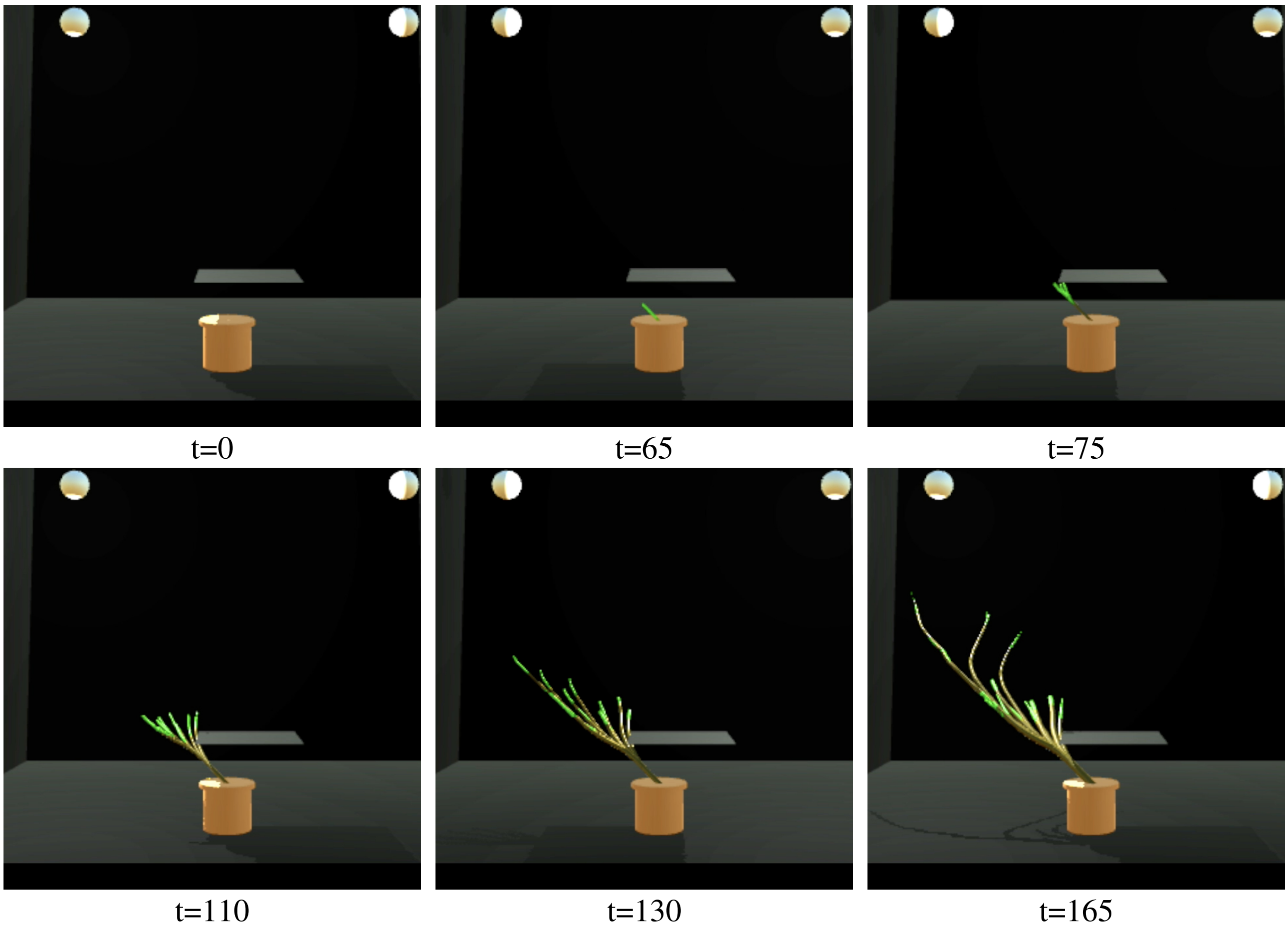


Figure 7. Screenshots of a swarm grammar specifically illustrating shadow avoidance. The plate hovering above the plant pot effectively shields the light from the top-right light source (depicted as a sphere). Avoiding the shadow, the plant's growth is dominated by a sidesways movement.

2.1.5 Interactivity

Currently, the interactive growth model incorporates several factors of plant behaviours including growth and branching, gravitropism, lignification, phototropism and shadow avoidance. There are other behavioural aspects that should be included as well, especially in the context of biohybrid applications. One such aspect would be creeping, e.g. to make effective use of any scaffolding machinery. Clearly the presented model has not been tailored to fit the behaviours and features of a specific biological model plant. We are currently investigating according approaches to automatically learn the model parameters from empirical data, as outlined by Wahby et al. (2015).

The advantage of the present model over other approaches lies in its interactivity. This means, due to its algorithmic simplicity, its incremental growth procedure and the underlying data model, it works in realtime. As a consequence, the model can be utilised in interactive simulations in which human users can seed plants, place obstacles, light sources, scaffolds or robots to tend the plants. We deem this a critical aspect of simulations of biohybrid systems due to their inherent complexity: The great numbers of interacting agents, their ability to self-organise and to reach complex system regimes, also due to constant interaction with the potentially dynamic and partially self-referential environment, makes it mandatory to develop a notion of a specific system's configuration's impact before deployment. In the following section, we present interfaces that can harness interactive simulation models for designing and planning biohybrid systems.

3 ROBOT GARDENS AR

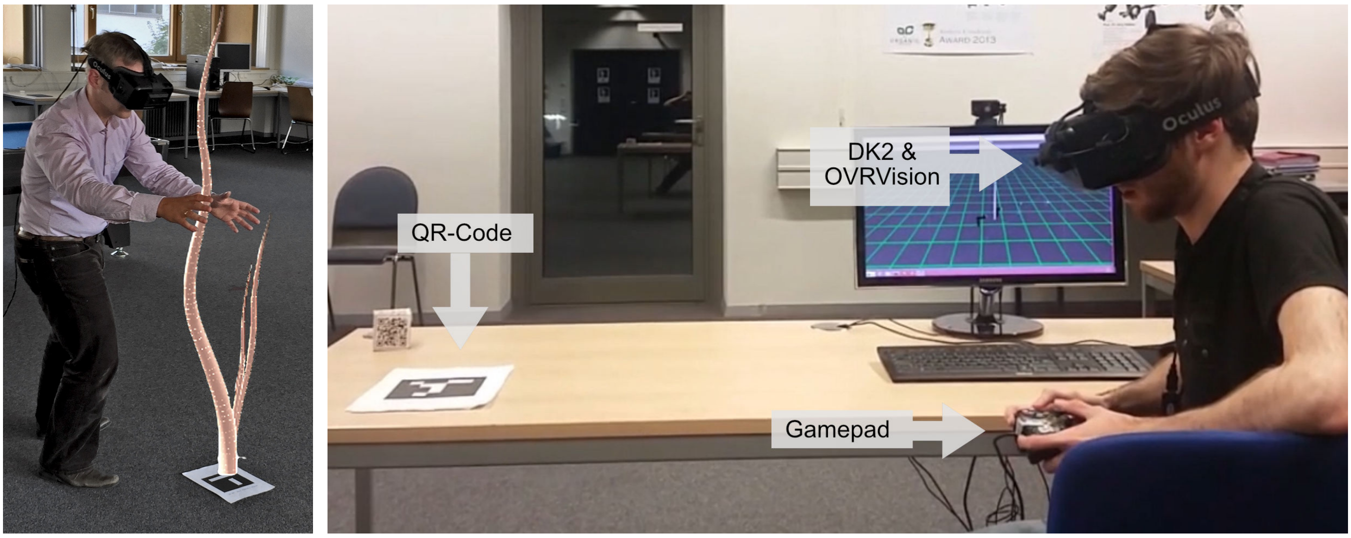
We previously presented an early prototype of an augmented reality system for designing and exploring biohybrid systems (von Mammen et al. (2016a)). We refer to the underlying concept as "robot gardens" as we envision the user to be immersed in the system of robots and biological organisms and be able to tend it like a gardener—not unlike the idea conveyed by von Mammen and Jacob (2007). Planning, planting and caring for the biohybrid "garden" has a lot in common with an actual garden, as it requires frequent attention over long time spans. Our envisioned interface for also cultivating robotic parts foresees to visually augment the objects with information about states and control programmes (von Mammen et al. (2016)). As implied in Figure 8(a), hand and finger tracking could render it feasible let the user interact with virtual organisms and mechanical parts like with physical object but also programmatically (Jacob et al. (2008)). In this section, we summarise our experiences with a first functional robot gardens prototype for augmented reality. It focusses on the technical feasibility and first analysis and evaluation of user interaction tasks.

3.1 Overview

Figure 8(b) captures the hardware setup of our first robot gardens prototype. It enhanced an Oculus DK2 virtual reality head-mounted display by means of a stereoscopic OVRVision USB camera. Information about a QR-code that appears in the video stream is extracted to maintain a point of reference with absolute coordinates. As its location and orientation relative to the user can be inferred from the QR-code image, arbitrary visual data can be overlaid on the video feed to augment reality. The user can introduce commands, for instance for placing and orienting robots or seeding plants by two means: First, the user's head orientation is tracked by the DK2 headset. The centre of the view can is utilised for selection or positioning tasks in the environment. Second, the user can select and execute individual commands such as pausing/playing or fast-forwarding the simulation or (de-)activating a robot by means of a gamepad.

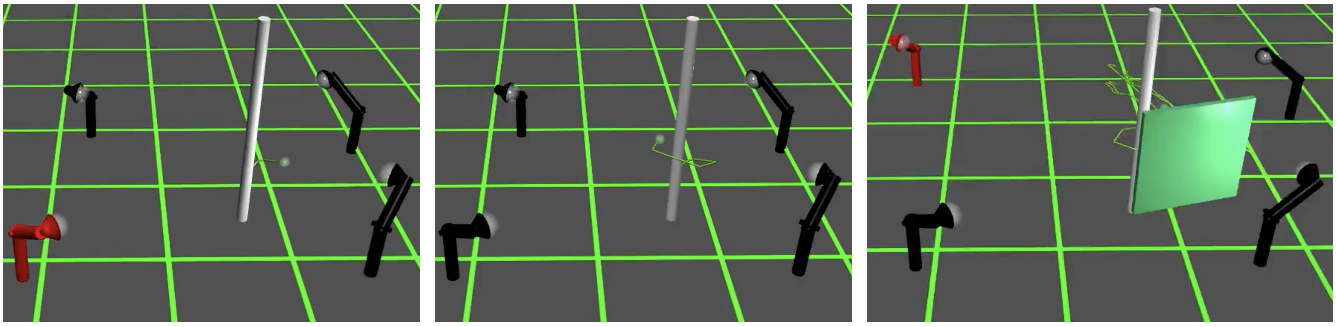
3.2 AR Session

Figure 9 shows the simulated content that is projected onto the video feed during simulation. In particular, one sees a pole at the centre of the screen. The user has placed four "lamp-bots", simple robots with a



(a) (b)

Figure 8. (a) Mock-up of an augmented reality situation where a plant-like structure grows from the floor that the user can interact with. (b) Overview of the robot gardens augmented reality prototype: A stereoscopic camera (OVRVision) extends the functionality of a head-mounted virtual reality display (an Oculus DK2). The camera feed is funnelled through to the DK2. Easily detectable QR-markers allow one to place virtual objects in space, at absolute coordinates. A gamepad acts as a simple control interface for the user.



(a) (b) (c)

Figure 9. (a) The red lamp-bot is highlighted in red as it is currently selected, ready for reconfiguration. (b) The plant-like structure grows around the pole as it is alternately attracted by one of the lamp-bots. (c) A panel has been placed to shield the plant physically from one light source.

spot-light mounted at the top. They are oriented towards the pole. In Figure 9(a), the bottom-left lamp-bot is shaded in red as it has just been placed by the user and it is still selected for further configurations, including its orientation. Figure 9(b) shows how the plant-like structure at the centre is growing around the pole due as it is only attracted by one lamp-bot at a time to describe a circular path. In Figure 9(c), the user has placed a panel between the bottom-right lamp-bot and the pole as to shield the lamp-bot's light from the plant-like structure but also to shield the plant-like structure physically from growing in this direction.

3.3 Usability

We conducted a short usability study to learn which aspects of our prototype work and which do not. The 12 students (22 to 25yrs, only two with a background in computer science or related fields) were introduced to the interface and then asked to achieve three tasks of increasing complexity. The first task was to merely place a lamp-bot within a given region. The second one required the user to orientate the lamp-bot to face a certain direction. As their third task, they needed to guide a phototropic and gravitropic growing plant-like structure around a pole—utilising panels, lamp-bots and configuring them. The biohybrid configuration depicted in Figure 9 is similar to some of the results created by the testers. We drew the following conclusion from this short study (detailed in: von Mammen et al. (2016a)).

1. Planning and designing biohybrid systems in augmented reality is an obvious approach and easily achievable.
2. The interactions among the biohybrid agents should be visualised, e.g. the light cones of the spot-lights.
3. A tethered hardware setup poses an unwelcome challenge, even under laboratory conditions.
4. In order to provide the means of complex configurations, naturalness of the user interface needs to be increased further: Gaze-based selection and positioning worked well but using a gamepad put a great cognitive load on most testers.
5. The interface can be improved by revealing addition information such as occluded lamp-bots (e.g. by rendering the ones above semi-transparently).
6. The visualisation can be further improved by mapping the actual lighting conditions onto the virtual, augmented objects.

4 VIRTUAL REALITIES

Our experiments confirmed that the concept of designing biohybrid systems by means of AR represents a viable approach. They also stressed that the shortcomings of tethered hardware solutions render more rigorous testing and development of such a system challenging. We are aware that recent advances in augmented reality hardware have already demonstrated that these teething troubles will soon be overcome, at affordable prices and providing reasonable processing power. However, instead of iteratively refining the augmented reality prototype, the goal to explore the design spaces of biohybrid systems can be achieved faster by means of virtual reality. Therefore, in order to explore novel spaces that opened up based on biohybrid design concepts, we decided to flesh out an according VR approach. In particular, architects involved in biohybrid research (see for instance Heinrich et al. (2016)) have been investigating the idea of braided structures as they are lightweight yet strong and structurally flexible—which are important properties when aiming at results from plant-robot societies. Accordingly, for a simulation-driven virtual reality world, we adapted swarm grammar agents to braid in a self-organised manner based on cues in the local environments. We do not decide whether the plants or the robots will eventually play the role of the braiding agents or how they will be realised technically. Rather, we assume that the designer of the future will have braiding agents available and he can deploy them at will. In this section, we briefly introduce this VR system.

4.1 VR Interface

Different from AR, where the user's natural environment is augmented by additional information—such as the information about the configuration and evolution of biohybrid systems over time—VR immerses the user into a virtual world, where even the surroundings can be of artificial origin. The greater the quality, the more natural the interactions and the faster its response, the more VR technology vanishes into the background. We say the user is more immersed, and, based on his emotional engagement, he can find himself fully present in VR (Slater and Wilbur (1997)). For the purpose of our experiment, we focussed on providing the functionality needed to quickly prototype certain physical, static environments, to place and, in parts, direct braiding agents. Figure 10(a) shows the VR gear comprised of one head-mounted display (HMD) and two 3D controllers. The HMD is tethered to a powerful desktop computer. The positional tracking information of the devices is calculated based on the two additional light-house boxes that shine

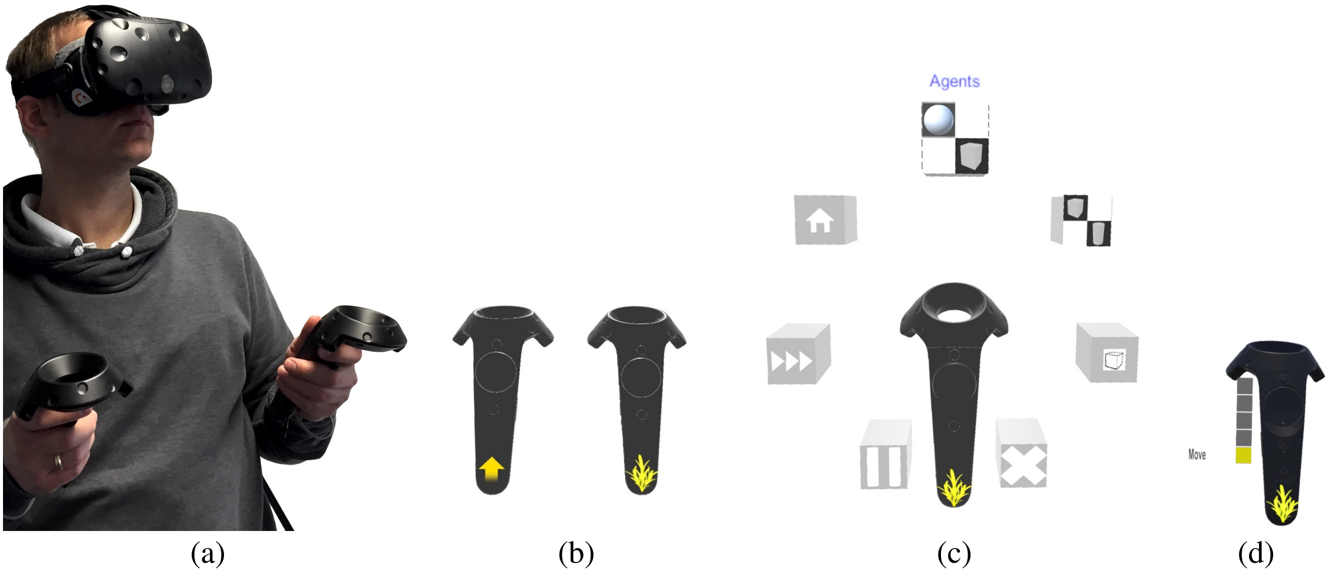


Figure 10. (a) .

from two diagonal ends of an approximately 3x3m^2 large area. In VR, the two controllers are displayed exactly where the user would expect them and they are augmented with two small icons as to distinguish their input functionality (Figure 10(b)). The arrow icon indicates that this controller is used to move the user through the virtual space: Pressing the flat round touch-sensitive button on the controller, an arc protrudes from the controller, intersecting with the ground. Accordingly, depending on the controller's direction and pitch, a close-by location on the ground is selected. When the user releases the button, his view is moved to this new location instantaneously. This approach to navigation in VR has been widely adopted as any animations of movement which do not correspond to one's actual acceleration may contribute to motion sickness (von Mammen et al. (2016b)). Options to control the simulation as well as any manipulations of the environment are made available by the second controller marked with a plant-like yellow icon. If the user presses the small round button above the big round touch-sensitive field, a radial menu as seen in Figure 10(c) opens up round the controller. The top-centred menu item is selected if the user presses the trigger button with his the index finger at the back of the 3D controller. The radial menu is rotated to the right or left by the corresponding inputs on the controller's touch-sensitive field. In order to quickly setup and assay environment, objects can be scaled or moved by means of the touch-sensitive field as well, if switched into the according transform mode as shown in Figure 10(d).

4.2 Braiding Agents

Braids are comprised of multiple threads that are pairwise interwoven. In order to create such a structure, a simple algorithm can be formulated, where a specific thread is identified based on its relative position to its neighbour threads and folded to cross them. Figure 11(a) shows our first approach to retrace such a centralised algorithm. In an open, biohybrid system, the agents—whether robots or biological organisms—need to act autonomously and in a self-organised fashion. Therefore, we created an according behavioural description that can be performed by each agent individually and globally results in a braided structure. In Figure 11(b), the latter, self-organised approach is shown in the context of two braiding swarm grammar agents: If a neighbour is close enough, they start rotating around the axis between the two. Braids across several threads can, for instance, be achieved by (1) expanding the agents' field of view to increase the probability to see multiple peers, (2) let the one with close-by neighbours but otherwise furthest away move

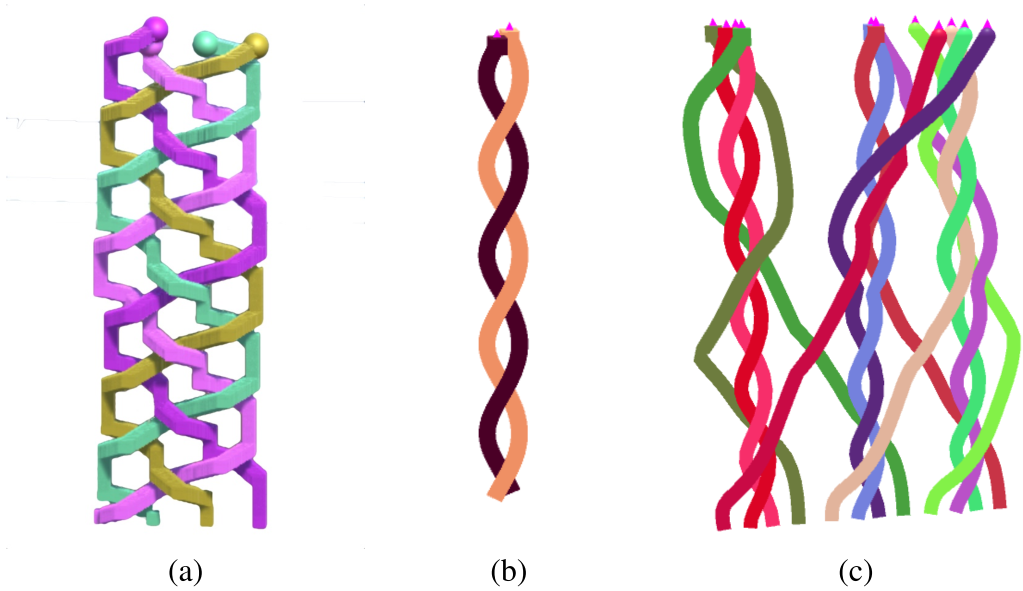


Figure 11. (a) Three threads are braided from the bottom upwards. Similar to a loom, one programme concerts the exact paths. (b) Two swarm grammar agents (pink pyramids at the top) see each other and make sure to cross each other's path in order to yield braided traces. (c) Twelve swarm grammar agents braiding together. As the parameter values are not properly adjusted yet, wider streaks seemingly occur at random.

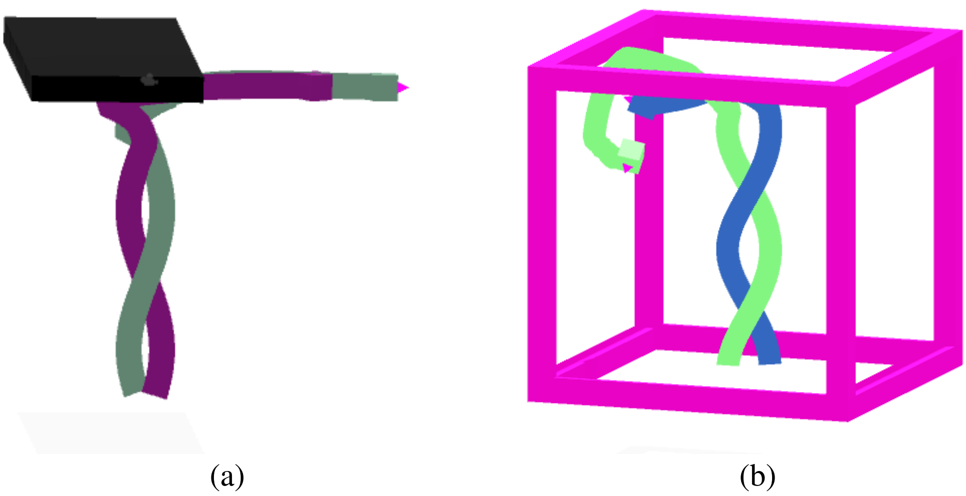


Figure 12. (a) The SG braid agents react to their physical environment and deflect from a plate. (b) Two SG braid agents are trapped inside a braiding volume which can be used to guide their evolution.

towards the opposite end of the flock while the others continue as they were. As this multi-agent braid takes considerable effort in terms of velocity regulation and parameter calibration as seen in Figure 11(c), we relied on a two-agent braiding function for our early experiments. In order to guide the braid agents, the user can place objects such as the plate in Figure 12(a) in VR space, which lets the agents deflect. Alternatively, as shown in Figure 12(b), so-called braiding volumes (in analogy to "breeding volumes" used by von Mammen and Jacob (2007)) can be deployed to enclose agents within specific spaces. These braiding volumes can be placed seamlessly as seen in the next paragraphs to provide arbitrary target spaces.

4.3 Braiding Experiments

In a first set of experiments, we asked students with computer science background as well as architecture students to test the VR braiding prototype and let their creativity roam freely, after a tutorial-based or oral, hands-on introduction to the interface and the simulation mechanics. We are still running the evaluations but one result is already emerging. It took the testers very little time (roughly 2 to 3 minutes) to familiarise themselves with the various aspects of the user interface. Therefore, we assume that it offered a very shallow learning curve despite the inherent complexity of the combined task of navigation, transformation, placement and simulation control. We could also learn that both student groups were intrigued by the autonomy of the braiding agents but due to a lack of explanation, they could not fully retrace the individual agents' behaviours in different situations. Clearly this is one of the aspects we aim at working next. Figure 13 shows examples of braided structures that were captured during the experiments. During the subsequent interviews, especially the architecture students and one architecture professor stressed that they foresaw great potential for biohybrid systems in design and construction and that they are convinced that research towards according simulations and user interfaces is crucial for its realisation.

5 CONCLUSION

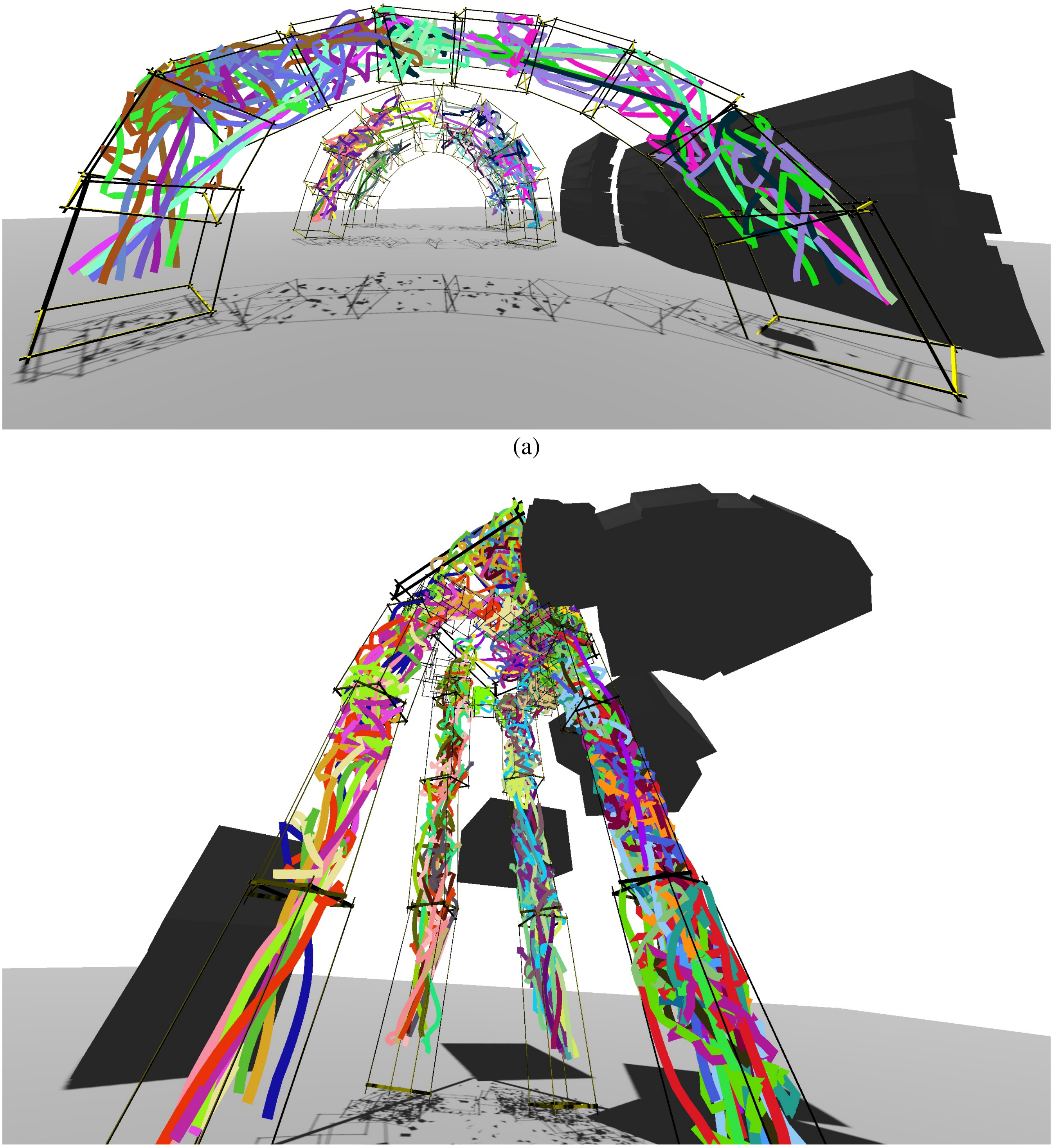
Biohybrid systems promise to act as enzymes, accelerating and automating various interaction cycles with nature that had previously been performed by humans. Considering time scales and the spatial distribution of robotic nodes, it is evident that biohybrid systems also have the potential to bring about completely new situations and artefacts. Simple biohybrid systems can be setup rather quickly, whereas their impact might not be evident right away. Accordingly, even for simple use cases, we are convinced that simulations should be queried to design and refine biohybrid systems. In order to render such simulations effective and accessibly support the user, they need to run at realtime speed, they need to deploy developmental data structures that grow over time and models that can be easily influenced by different kinds of interactions.

In this article we motivated agent-based procedural content generation in the context of biohybrid simulations due to its inherent compatibility with interactive simulations. We revisited swarm grammars as a means to combine developmental processes and agent-based models. In order to use swarm grammars to drive basic plant models in interactive biohybrid simulations, we extended the previously existing models with basic botanical behavioural patterns including gravitropism, phototropism, growth, lignification and shadow avoidance. In order to realise these model extensions, we established a hierarchical representation of the swarm grammars' trace segments. This data structure serves as a simple means to regulate the metabolic factors during plant growth, at the same time. We showed how an augmented reality setup can deploy interactive models to inform designers during the planning stage of biohybrid simulations, and finally, we introduced a virtual reality system to explore novel design spaces that will be opening up due to biohybrid systems research and development.

Like other in all other interactive simulations, the quest for of biohybrid systems is multi-facetted and, therefore, challenging. User interface designs and technologies can equally fast let the user experience deteriorate as can insufficient models or slow simulation speeds. Therefore, all the aspects presented in this article are tightly interwoven and, at the same time, all deserve more work to support biohybrid systems design and dissemination. In order to accelerate these efforts, integrated iterative research and development cycles should be established to, for instance, seamlessly integrate new mechanical and logical capabilities of robots for biohybrid systems, or to improve the predictability and accuracy of realtime plant models.

REFERENCES

Cristea, A. and Liarokapis, F. (2015). Fractal nature - generating realistic terrains for games. In *2015 7th International Conference on Games and Virtual Worlds for Serious Applications (VS-Games)*. 1–8.



(a)

(b)

Figure 13. Braiding volumes are arranged as (a) arcs and (b) columns to guide the building swarms of agents.

doi:10.1109/VS-GAMES.2015.7295776

Culik, K. and Lindenmayer, A. (1976). Parallel graph generating and graph recurrence systems for multicellular development. *International Journal of General Systems* 3, 53–66 de Campos, L. M. L., de Oliveira, R. C. L., and Roisenberg, M. (2015). Evolving artificial neural networks through l-system and evolutionary computation. In *2015 International Joint Conference on Neural Networks (IJCNN)*. 1–9. doi:10.1109/IJCNN.2015.7280535

Fiscus, E. L. (1975). The interaction between osmotic-and pressure-induced water flow in plant roots. *Plant Physiology* 55, 917–922

Hamon, L., Richard, E., Richard, P., Boumaza, R., and Ferrier, J.-L. (2012). Rtil-system: a real-time interactive l-system for 3d interactions with virtual plants. *Virtual Reality* 16, 151–160

Heinrich, M. K., Wahby, M., Soorati, M. D., Hofstadler, D. N., Zahadat, P., Ayres, P., et al. (2016). Self-organized construction with continuous building material: Higher flexibility based on braided structures. In *Foundations and Applications of Self* Systems, IEEE International Workshops on* (IEEE), 154–159

Jacob, R. J., Girouard, A., Hirshfield, L. M., Horn, M. S., Shaer, O., Solovey, E. T., et al. (2008). Reality-based interaction: a framework for post-wimp interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems* (ACM), 201–210

Kniemeyer, O., Barczik, G., Hemmerling, R., and Kurth, W. (2008). Relational Growth Grammars—A Parallel Graph Transformation Approach with Applications in Biology and Architecture. In *Applications of Graph Transformations with Industrial Relevance: Third International Symposium, AGTIVE 2007, Kassel, Germany, October 10-12, 2007, Revised Selected and Invited Papers* (Berlin, Heidelberg: Springer-Verlag), 152–167

Kniemeyer, O., Buck-Sorlin, G., and Kurth, W. (2006). Groimp as a platform for functional-structural modelling of plants. In *Functional-Structural Plant Modelling in Crop Production*, eds. J. Vos, L. F. M. Marcelis, P. H. B. deVisser, P. C. Struik, and J. B. Evers (Springer), 43–52

Kniemeyer, O., Buck-Sorlin, G. H., and Kurth, W. (2004). A graph grammar approach to artificial life. *Artificial Life* 10, 413–431

Lindenmayer, A. (1971). Developmental systems without cellular interactions, their languages and grammars. *Journal of Theoretical Biology* 30, 455–484

Onishi, K., Hasuike, S., Kitamura, Y., and Kishino, F. (2003). Interactive modeling of trees by using growth simulation. In *Proceedings of the ACM symposium on Virtual reality software and technology* (ACM), 66–72

Prusinkiewicz, P. and Hanan, J. (2013). *Lindenmayer systems, fractals, and plants*, vol. 79 (Springer Science & Business Media)

Prusinkiewicz, P. and Lindenmayer, A. (1996). *The Algorithmic Beauty of Plants* (Springer-Verlag)

Rankin, J. (2015). The uplift model terrain generator. *International Journal of Computer Graphics & Animation* 5, 1

Reynolds, C. W. (1987). Flocks, herds, and schools: A distributed behavioral model. *Computer Graphics* 21, 25–34

Shaker, N., Togelius, J., and Nelson, M. (2014). Procedural content generation in games

Slater, M. and Wilbur, S. (1997). A framework for immersive virtual environments (five): Speculations on the role of presence in virtual environments. *Presence: Teleoperators and virtual environments* 6, 603–616

Stern, K. R., Jansky, S., and Bidlack, J. E. (2003). *Introductory plant biology* (McGraw-Hill New York)

Thompson, D. (2008). *On Growth and Form* (Cambridge, UK: Cambridge University Press)

Tyrrell, A. M. and Trefzer, M. A. (2015). *Evolution, Development and Evolvable Hardware* (Berlin, Heidelberg: Springer Berlin Heidelberg). 3–25. doi:10.1007/978-3-662-44616-4_1 von Mammen, S. (2006). *Swarm Grammars - A New Approach to Dynamic Growth*. Technical report, University of Calgary, Calgary, Canada

von Mammen, S. and Edenhofer, S. (2014). Swarm grammars gd: Interactive exploration of swarm dynamics and structural development. In *ALIFE 14: The International Conference on the Synthesis and Simulation of Living Systems*, ed. H. L. et al. ACM (MIT press), 312–320

von Mammen, S., Hamann, H., and Heider, M. (2016a). Robot gardens: An augmented reality prototype for plant-robot biohybrid systems. In *Proceedings of the 22Nd ACM Conference on Virtual Reality Software and Technology* (New York, NY, USA: ACM), VRST '16, 139–142. doi:10.1145/2993369.2993400

von Mammen, S. and Jacob, C. (2007). Genetic swarm grammar programming: Ecological breeding like a gardener. In *CEC 2007, IEEE Congress on Evolutionary Computation*, eds. D. Srinivasan and L. Wang (Singapore), IEEE Press, 851–858

von Mammen, S. and Jacob, C. (2009). The evolution of swarm grammars: Growing trees, crafting art and bottom-up design. *IEEE Computational Intelligence Magazine* 4, 10–19

von Mammen, S., Knote, A., and Edenhofer, S. (2016b). Cyber sick but still having fun. In *Proceedings of the 22nd ACM Conference on Virtual Reality Software and Technology* (ACM), 325–326

von Mammen, S., Schellmoser, S., Jacob, C., and Hähner, J. (2016). *The Digital Patient: Advancing Medical Research, Education, and Practice* (Hoboken, New Jersey: John Wiley & Sons), chap. 11. Modelling & Understanding the Human Body with Swarmscript. Wiley Series in Modeling and Simulation. 149–170

von Mammen, S. and Steghöfer, J.-P. (2014). *The Computer after Me: Awareness and Self-Awareness in Autonomic Systems* (World Scientific Press), chap. Bring it on, Complexity! Present and future of self-organising middle-out abstraction. 83–102

Wahby, M., Divband Soorati, M., von Mammen, S., and Hamann, H. (2015). Evolution of controllers for robot-plant bio-hybdrids: A simple case study using a model of plant growth and motion. In *Proceedings of 25. Workshop Computational Intelligence* (Dortmund: KIT Scientific Publishing), 67–86

Wooldridge, M. (2009). *An introduction to multiagent systems* (Wiley), 2nd edition edn.

Wyman, C., Hoetzlein, R., and Lefohn, A. (2016). Frustum-traced irregular z-buffers: fast, sub-pixel accurate hard shadows. *IEEE transactions on visualization and computer graphics* 22, 2249–2261