



HAL
open science

Immersive and interactive visualisation of a virtual honey bee colony

Thomas Alves, Jérémy Rivière, Vincent Rodin, Thierry Duval

► **To cite this version:**

Thomas Alves, Jérémy Rivière, Vincent Rodin, Thierry Duval. Immersive and interactive visualisation of a virtual honey bee colony. EuroVR 2019: International Conference on Virtual Reality and Mixed Reality, Oct 2019, Tallinn, Estonia. 10.1007/978-3-030-31908-3_23 . hal-02297300

HAL Id: hal-02297300

<https://hal.science/hal-02297300>

Submitted on 26 Sep 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Immersive and interactive visualisation of a virtual honey bee colony

Thomas Alves¹, Jérémy Rivière¹, Vincent Rodin¹, Thierry Duval²

¹ Univ Brest, Lab-STICC, CNRS, UMR 6285, Brest, France
{firstname.surname}@univ-brest.fr

² IMT Atlantique, Lab-STICC, CNRS, UMR 6285, Brest, France
thierry.duval@imt-atlantique.fr

Abstract. Social insects and more specifically honey bees have very complex, powerful and interesting task allocation abilities. They are able to distribute their workforce effectively without any central control, using simple mechanisms based on stimuli (dances, pheromones), interactions, thresholds and feedback loops. Self-organisational concepts and some of those mechanisms, like pheromones, are invisible to the human eye. In order to help the user grasp the complexity of the task allocation, we propose to make them visible in a virtual, immersive and interactive environment. First, this implies that the system must be able to simulate and display in real-time around 30 000 bees interacting with each other, emitting clouds of pheromones. Secondly, this implies also that the user should be able to alter in real-time the environment of the bees (e.g. by manipulating the frame of the hive) and visualise the effects on the organisation, potentially days later. Finally, we would like to give to biologists and beekeepers some domain-related, intuitive and natural ways of interacting with the hive. We describe in this article these issues in more details, and how we plan to tackle them. This is a “work in progress”, therefore a lot of work has still to be done, mostly surveying and modelling the interactions.

Keywords: Complexity · Agent-based simulation · Tangible interfaces · Immersive environment · Training and education

1 Introduction

Social insects are well known for their robust and effective decentralised organisation. They are able to distribute their workforce amongst all the different tasks to be done, without any global knowledge of the state of the colony. This task allocation ability [1] lies on stimuli (dances, pheromones), interactions between individuals, internal thresholds and positive and negative feedback loops.

The honey bee *Apis Mellifera*, the most common honey bee in Europe, uses these mechanisms in a variety of ways, such as dances to recruit their nest-mates to foraging activities or many different pheromones to inhibit or favour certain behaviour [9, 8, 4]. The first part of our project is to model these mechanisms, within an agent-based simulation [7, 1]. Agent-based modelling is a bottom-up approach that focuses on individuals, their behaviour and interactions (at the micro-level), to deduce the overall behaviour of

the system (macro-level) by emergence. The agent-based approach is thus particularly suited to model and simulate complex systems. Following this approach, each honey-bee is an agent, described by its biological cycle, behaviour, and its interactions with its environment, including other agents. We work with biologists and beekeepers to help us build and validate the model, and make it as relevant as possible.

We intend to model most of the in-hive tasks (feeding the brood, cleaning the combs, foraging etc.), triggered (or inhibited) by different stimuli such as pheromones. Self-organisational concepts are driven by invisible mechanisms, and thus quite challenging to understand. There comes the second part of the project, a virtual, immersive and interactive simulation. With virtual reality we want to allow the user to comprehend those concepts, by showing the mechanisms behind them, hiding some parts of the hive, abstracting the bees to focus only on pheromones, i.e. enhancing the view.

Related works on honey bee colony visualisation aim mainly at help beekeepers make decisions, by providing them information about honey bee populations [2, 3, 6]. For example, the authors of [2] use Augmented Reality to display data from several sensors in a range of real hives and allow the user to go through all the data in an intuitive and immersive manner. These data are collected at the hive (macro) level, such as temperature and weight. They are thus able to know in which hive certain bees are and study “bee drift”: when bees from a colony leave and join another colony. We, on the other hand, work on a simulated honey bee colony and focus on the “micro” world, at the bee level. Our goal is not to (directly) help the decision making of beekeepers, but to allow them to visualize and understand their actions’ effects on the organisation of the (virtual) colony, inside the hive. Section 3.3 provides some examples of such beekeeping actions.

2 Scientific challenges

We highlight in this section three of the main issues that we face.

First, we want to visualise a very large system in 3D, with tens of thousands individuals, interacting in various ways with each other, and allow the user to interact in real-time with the system, changing the state of the simulation. The performance of such a system needs to be considered.

Secondly, the user should be able to interact with the hive, and watch the colony adapt to those actions in real-time, or fast forward to a later stage of the world to see how the colony handled (or not) the action’s effects in the long term. It is necessary to consider ways to handle those multiple time interactions, both on the simulator and the visualisation.

Finally, this simulator should be used by biologists and beekeepers in different uses: a scientific use, aimed at biologists, to evaluate the role of stimuli and other mechanisms on self-organisation; and educational use, to allow new and experienced beekeepers to learn good beekeeping practices through predefined scenarios and realistic means of interaction. To do so, we would like to implement domain-related, intuitive and natural ways of interacting with the hive.

3 Preliminary Work

3.1 Visualisation and Interactions Challenges

We describe here our work on the first two challenges described previously.

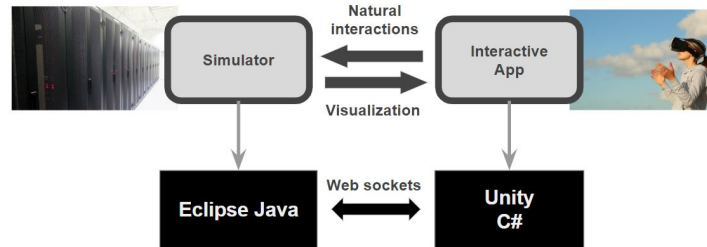


Fig. 1. Decoupled architecture. A java server holding the simulation and the interactive and immersive app in Unity3D will talk to each other and synchronise through Web sockets.

We propose to separate the simulation and its visualisation, following the advice of [5], by using a Java server calculator to run the model, and a Unity3D client for visualisation and interactions (see Figure 1). The first major effect of decoupling the calculator that holds the simulation, and the interacting app, is that we can model each part without the constraint of the other part. The simulator won't have to compute any complex visualisation nor to draw it onscreen in real-time, and the interactive system won't have to bother with the calculation of the agent-based system. We may also gain some performance, as we can easily run those two parts on separate computers: the authors of [5] have shown that their decoupling does not create any performance bottleneck. Moreover, by creating a "Rest API-like" server, we set the server free from the graphical client's updates, and even allow ourselves to change the interactive app later on if needed, without changing the inner behaviour of the simulation.

Another effect of this separation is the decoupling of the time between the simulator (virtual time) and the interacting app (in real-time to allow the user to interact with the frames and the colony). This is a disadvantage, as we will have to ensure that the simulator and the interacting app stay synchronised: an action performed at time t by the user must occur in the simulation at the same time. On the other hand, we need this decoupling because we intend to give the possibility to the user to fast forward in a later time to observe the effects of an action, so the simulator must be able to compute the simulation very fast. Moreover, those two highly different speeds will require us to implement a way for the user to understand the time flow and interact with it.

3.2 The Virtual Environment

The environment consists of the Dadant Hive, the standard beehive for honey harvesting. The hive is mainly a wooden box, with rails on the sides to guide a few frames

inside. We modelled it in Unity3D with the hexagonal cells in the frames, which are where the larvae are raised and the resources packed. Figure 2 shows an example of such a visualisation, in which the server randomly filled all the cells, then sent the data to the client. Inside Unity3D, we can interact with these filled frames, making them rise so we can see their content. This will be the basis for our work.

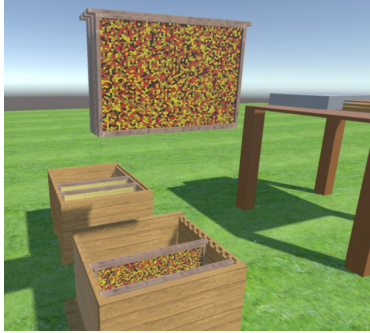


Fig. 2. A Unity3D simulation that visualises the results of a Java simulation.

Our virtual world, with the hive, will be filled with honey bees. Typically, in Unity3D, we would instantiate each individual as a “GameObject”, and let Unity3D handle the movements and display. But as honey bee colonies can count up to 50 000 individuals, that many GameObjects would cause severe frame rate drop and ruin the immersive experience.

Unity3D is way more effective iterating inside a GameObject as it is to iterate through them. So, instead of having a few thousand GameObject, we should have a single one, but containing a few thousands of shapes scattered around the world. We will, therefore, use a point cloud inside Unity3D, and Unity3D’s implementation of a point cloud is a very good start. We managed to display a few hundred thousands points with a very decent frame rate, around 60fps. We can now

use the same idea to represent different layers of the world, having, for example, a Point-Cloud object representing all the bees, and another one for the pheromones (or different objects for different types of pheromones). In section , we discuss how the visualisation of the organisation could look like.

3.3 Natural Intuitive Interactions

We also worked with beekeepers and bee biologists to know how they interact with their real hives. We recorded a dozen of interactions. This will be the basis of our model, making it as relevant as possible. We also recorded their expected effects on the organisation of the colony and will use these data to validate our task allocation model and the interaction model.

For example, a beekeeper sometimes has to remove the queen of a colony, often to replace it with a younger one. This change usually increases the number of eggs laid, which will require much more nurses, food and space. Beekeepers can also change the order of the frames by performing a circular permutation of all the frames. That movement changes drastically the environment of the colony and requires a self-adaptation. The last example is the more extreme and consists of the beekeeper removing all the brood of the colony, to prevent a disease to spread. This is done by renewing all the frames of the hive and has a huge impact on the colony.

In a nutshell, each beekeeper move has a significant impact on multiple feedback loops, and, in extent, on the task allocation of the whole colony. We discuss in the next section the interaction means that could implement these actions.

4 Future Work and Perspectives

Different ways of interacting could help us answer to the issues listed in section 2. Natural and intuitive interactions could be implemented thanks to a virtual hive that can be manipulated with tangible interactors. For example, a common beekeeper action is to remove a frame from the hive and then hit it on the top to make all the bees on it fall, allowing a better and safer work on the frame. This simple, key movement would highly benefit a tangible interactor, to reproduce at best the real feeling of the movement, enhancing the immersion feeling and maybe enhance the learning experience of the user as well. A solution could either be a real tracked Dadant hive with real tracked frames, or a re-built simplified version with only the structural needs, as it does not have to look like a hive. We still have to study the feasibility of both ideas.

The vast majority of the beekeeping interaction with the hive requires the handling of a frame, enhancing the needs for tangible interactors. For example, the queen removal action described in the previous section requires to remove a frame from the hive, manipulate it to find the queen, then lift it with an interactor, insert the new queen and then put the frame back into the hive. In the same way, a rotation, circular permutation of all the frames requires to lift a frame, remove the bees on it by hitting it on the top, store it in a dedicated spot near the user, move all the other frames, and then put the stored frame back on the other side of the hive.

Biologists and beekeepers won't have the same needs regarding the interactions and the visualisation of the system, even if they globally interact in the same way with the frames. That's why we intend to implement a system of layers, allowing each user to effectively chose what they want to see. For example, a biologist studying the impact of pheromones on the colony organisation could decide to hide the bees, the frames and the hive, and only see the pheromones. We could even decide to hide or show bees based on their activities, or age, and then display their perception.

It's not clear yet, however, how the organisation will be shown to the user. Bringing that abstract concept on the visible layer will require further work, but a first idea would be to create geometries just above each agent, describing what they perceive (with a bar chart) and what they do (for example a square floating above a bee would mean that it's currently a nurse). We could even select an agent and then see its whole history of perception and action with graphs displayed inside the virtual world.

To allow the user to navigate in time, a tangible time controller is considered, which should allow the user to fast forward and see the state of the colony in the future. We are thinking about possible time travel to the past, but we are still unsure if this would be meaningful for the project. Further discussions with biologists and beekeepers are planned.

Finally, a potential multi-verse metaphor could be useful for the user to compare the consequences of multiple gestures upon the same colony. The user would be able at any time to create a new universe, holding a copy of the current state of the colony. Then, he could perform any interaction with the hives in any verse. The verses would all be independent and thus, multiple hives could be watched at the same time. We could, for example, have a “default colony” in the first verse, and never interact with it. Then effectively compare it to the other verses where we would have performed many changes.

5 Reference to the official publication

Springer Nature Switzerland AG 2019

P. Bourdot et al. (Eds.): EuroVR 2019, LNCS 11883, pp. 1–18, 2019.

The final authenticated version is available online at:

https://doi.org/10.1007/978-3-030-31908-3_23

References

1. Bonabeau, E., Dorigo, M., Theraulaz, G.: From Natural to Artificial Swarm Intelligence. Oxford University Press, New York, NY, USA (1999)
2. Engelke, U., Marendy, P., Susanto, F., Williams, R., Mahub, S., Nguyen, H., d. Souza, P.: A visual analytics framework to study honey bee behaviour. In: IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS). pp. 1504–1511 (2016)
3. Engelke, U., Hutson, H., Nguyen, H., de Souza, P.: MelissaR: Towards augmented visual analytics of honey bee behaviour. In: Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems. pp. 2057–2063. CHI EA '16, ACM, New York, NY, USA
4. Le Conte, Y., Mohammadi, A., Robinson, G.E.: Primer effects of a brood pheromone on honeybee behavioural development. Proceedings of the Royal Society of London. Series B: Biological Sciences **268**(1463), 163–168 (2001)
5. Louloudi, A., Klügl, F.: A new framework for coupling agent-based simulation and immersive visualisation. Proceedings - 26th European Conference on Modelling and Simulation, ECMS 2012 (05 2012)
6. Nguyen, H., Ketchell, S., Engelke, U., Thomas, B.H., de Souza, P.: Augmented reality based bee drift analysis: A user study. In: 2017 International Symposium on Big Data Visual Analytics (BDVA). pp. 1–8 (2017)
7. Schmickl, T., Crailsheim, K.: Analysing honeybees' division of labour in broodcare by a multi-agent model p. 9 (2008)
8. Schmickl, T., Karsai, I.: Integral feedback control is at the core of task allocation and resilience of insect societies. Proceedings of the National Academy of Sciences **115**(52), 13180–13185 (2018)
9. Seeley, T.D.: The Wisdom of the Hive: the social physiology of honey bee colonies. Harvard University Press, Cambridge, Mass (1995)