An Overview of Simulation Software for Non-Experts to Perform Multi-Robot Experiments

Nur Raihan Ramli[1], Sazalinsyah Razali[2]
Robotics and Industrial Automation Research Group
Center of Excellence for Robotics and Industrial Automation
Universiti Teknikal Malaysia Melaka
76100 Durian Tunggal, Melaka, Malaysia
Email: [1]nurraihanramli@gmail.com, [2]sazalinsyah@utem.edu.my

Mashanum Osman
Creative Media Lab
Center for Advanced Computing Technologies
Universiti Teknikal Malaysia Melaka
76100 Durian Tunggal, Melaka, Malaysia
Email: mashanum@utem.edu.my

***Abstract*—Simulation is an initial approach to determine the experiment feasibility, especially for a complex robotics environment. This paper give an overview of five simulation software for non-expert developers to quickly perform the multi-robots simulation. In the advanced robotics field, further research requires simulations, especially which involve multi-robots as to run quickly. There are many robotics simulation software exist. In this paper, we focus only on five simulation software, namely NetLogo, GAMA Platform, Webots, Player/Stage and V-REP. To distinguish the differences between this five tools, we made a comparison of general information and other criteria. Then, we rated them based on the criteria and user necessity. Therefore, the users can easily know which software is suitable for them to use.**

Keywords—multi-robot systems; overview; simulation software; non-experts; GAMA Platform

I. Introduction

The simulation experiment is crucial as it is a strategic method in all field, particularly in robotics field [1]. The presence of robotics simulation tools as a platform for featuring the virtual robot helps the researcher and developer for presenting the prototype of the real robot. Simulation software act as a medium to instantiate and define the robustness of the novel algorithm and concurrently, it can reduce the development time and cost of multi-robots. It is essential to test the robustness of robots on the simulation before implement it to the real robot. For instance, the development of swarm robotics using simulation software to some extent helps the researcher and developer to discover new approaches instead of facing with hardware failures repeatedly. In other words, simulation software can aid in solving the complexity issues in robot simulation.

Nowadays, robotics mechanism is enhanced and applied in various technology to solve desired task. The emergence and innovation of simulation software need to be considered since the higher robots complexity required an advance terrains and functionalities [2]. The focus in this paper is the simulation software that is easier for the non-expert user to perform multi-robot experiments. The multi-robot here can be represented as agents or robot models. Most application domains widely use agent-based model approach, but its use is still restricted by the limitation of generic yet ready-to-use tools supporting the design and the simulation of complex models [3]. Therefore, various tools for modeling and simulation can be used, for instance, Swarm, Webots, NetLogo, Player/Stage [4], V-REP [5], and Gazebo.

II. Related Works

There is a variety of agent-based simulation, and robotics simulation software can be used to perform multi-robot experiments. For agent-based simulation, there are Repast, MASON, Swarm, and NetLogo. Instead of choosing Repast, MASON, and Swarm, we choose NetLogo as it is suitable for building a multi-agent complex environment. Besides that, Swarm is extremely not easy to install and run successfully, and it is only suitable for expert user [6]. Meanwhile, Repast requires strong computer science skills and it is very complex and tricky to use [2]. MASON can perform multi-robot simulation, but it is not a right choice for non-expert user, but suitable for experienced programmers [2]. Therefore, we decided to choose NetLogo and another agent-based simulation software named GAMA platform [7]. Both software meet the needs for performing multi-robot experiments. We will also discuss on robotics simulation software, namely Webots, Player/Stage and V-REP.

NetLogo is an easy-to-learn authoring environment tool that is exclusively for novice programmer since the platform allows the user to create their models without a deep knowledge in programming [8]. Also, NetLogo enables the user to model and simulate complex phenomena across a wide variety of environments for multi-robots. As for the GAMA platform, Macatulad and Blanco in [9] stated that GAMA 3D features allow for modeling not only for outdoor population spaces but also for the indoor spatial environment. It even allows for 3D visualizations of the agent simulation. Thus, this gives the advantage for the implementation of robotics simulation with the new surroundings even when using agent simulation platform.

III. Overview of Simulation Software

This section discusses each simulation software, in general, to give the idea especially for non-expert users to choose the right tools for the multi-robot experiment. There is various software for robots simulation in the evolution of robotics technology. Either robots simulation tools or agents simulation tools, both allow the user to run the algorithm. In our case, we decided to select five simulation software namely, Webots, NetLogo, Player/Stage, GAMA Platform and V-REP. We selected NetLogo because it is free and frequently used for multi-agents simulation. The simple features of NetLogo ease the user of running the simulation smoothly. Webots is a commercial software and fortunately, one of the research groups already own Webots license so that we could run the simulation without restriction. As one of our team members had used Player/Stage before, we want to explore this software in details. The research group also planned to buy another robotics simulation called V-REP. V-REP is seemingly much advanced as it comes with real environment models like trees and furniture. On the other hand, the application of GAMA platform as robots simulation tool is new and hard to find. Therefore, we select GAMA to reveal its usefulness in the robotics area. The justification of these five tools is due to the above-stated reasons.

A. Netlogo

NetLogo is an open-source software platform dedicated to agent-based simulation and modeling environment. Users can download NetLogo simulation software from NetLogo website [10]. A user can install NetLogo on any Java Virtual Machine (JVM) since it can operate on most computing platform such as Windows, Linux, and Mac operating system. NetLogo allows for complex modeling up to thousands of independent agents that can run simultaneously [8]. NetLogo represents its agents as turtles. NetLogo is not a robotics simulation software, and it could not specify the robot types in the modeling as it is an agent-based platform. Nevertheless, the user still can make a robot simulation as agents in NetLogo and even for multi-agents modeling.

Besides that, NetLogo is suitable not only for a beginner but non-expert users too. NetLogo itself is a computer language, and it is simple compared to other programming languages [11]. As NetLogo is designed for educational purposes, the user does not require much programming experience to simulate the model. NetLogo simple user interface [11] allow the user to manage the modeling through user interface elements like button or the command center area provided in the interface. Furthermore, NetLogo comprised of an enormous collection of pre-built simulations from various fields for the user to explore and to alter the models.

Other than that, NetLogo provides modeling tutorial as in [12] to guide the user. These features to some extent are very useful to give an initial knowledge and modeling experience to the user. It can also train the non-expert user to modify the existing model rather than creating from scratch. Although NetLogo is capable of 3D visualization, there are limitations to 3D NetLogo as it requires language extension and 3D graphics to make it easier to build [11].

B. GAMA Platform

GIS and Agent-based Modeling Architecture or simply known as GAMA is a Java open-source software. Users can download GAMA simulation software on the GAMA website [7]. Currently, GAMA website and repository are being moved to GitHub [13]. GAMA platform operates with GPL license and runs on Windows, Mac OS X or Linux operating system. As to complete GAMA installation and for better performance during simulation, the user needs to install JVM, a sort of JDK version.

GAMA platform is commonly used in the environmental sector application due to its specialty to integrate GIS with agent-based modeling and multi-scale control. As with NetLogo, GAMA performs the modeling and authoring environment with its syntax called GAMA Modeling Language (GAML). GAMA simulation engine able to provide the ability to support multi-level model [14].

The graphical user interface of GAMA is quite complex primarily for a first-time user. If user familiar with GAMA environment, it will be a lot easier to use GAMA. Like NetLogo, robot model is excluded in simulation libraries since GAMA is developed as an agent-based simulation. However, the user can perform the modeling by representing the robots as agents. This way is much easier for non-expert user instead of manipulating the codes for specific robot modeling.

Moreover, GAMA simulation software provides various models libraries for the user to use or alter. Another special feature of the GAMA platform is it comes with provided agent behaviors (called skills in GAMA). This feature allows users to call their functions or even enable advanced users to create their own skills into their codes. Finally, the user can get a complete documentation at the GAMA platform website [7] for references.

C. Webots

Unlike NetLogo and GAMA platform, Webots [15] is a commercial robotics simulation software that widely used in the robotics area. Webots can run on multi-platform computing such as Windows, Mac OS X, and Linux. Webots is proprietary software and only allow for a 30 days free trial. This restriction can interfere the user's interest to keep using this software particularly for those who are new to robotics world. Despite that, it is undoubted that Webots has been utilized as the simulation platform for most robotics applications [16], [17].

Webots functioned well as common robotics simulation tools with its rapid prototyping environment [18]. To use Webots, users need to know the basic programming in C, C++, and Java. Other programming languages that can be used for Webots are Matlab and Python. In terms of functionality, it seems slightly useful particularly for complex models since Webots provide accurate physics simulation. Also, users can use the existing robot models and altering the script according to the requirements. In addition, Webots not only complete with robots models, but there are complete libraries for sensors and actuators available for users to individually plug them in robot model and tune it [19].

Furthermore, the simulation in Webots can run faster even in a complex environment and large worlds by using virtual time. These features allow the user to see the movement of the simulation precisely, and this method obviously save the time comparing to run in real robots.

D. Player/Stage

Player/Stage is a robotic simulation software that comprises of two separate components [20], [21]. The Player component is the server that can be run on several robot hardware.

Meanwhile, Stage is the interface for Player server, i.e. the client. Stage can also run as a standalone simulation tool but will not be directly transferable to robot hardware. The project has another component, called Gazebo that is a 3D robot simulation platform [22]. The Player/Stage platform is excellent if one wish to be able to transfer simulations onto real robots.

In addition, the platform is open source, meaning it is free to download and use. However, the learning curve to using it and the complexity is quite high. It is confusing for the non-expert user to handle Player/Stage software because the installation steps are quite complex for beginners, and there is no specific user interface that show the simulation steps. Moreover, it currently runs on Linux-based machines. Other than that, full and proper documentation is quite difficult to get.

E. Virtual Robot Experimentation Platform

The last simulation tool presented in this paper is Virtual Robot Experimentation Platform (V-REP). It is a freeware and downloadable. In contrast with other simulation software, V-REP comes with unlimited Educational version that operates on GNU GPL license and V-REP commercial Evaluation version. V-REP able to perform as a stand-alone application and can also integrate with other application easily [23].

V-REP main script is LUA script. However, it allows the user to select among the various programming techniques concurrently. That means, there are also other supported programming languages such as C/C++, Python, Java, Matlab, Urbi, and Octave. Nevertheless, it is challenging to understand V-REP stream, especially for the novice user.

The programming techniques provided by V-REP are embedded scripts, add-ons, plug-ins, remote API clients and ROS nodes [24]. V-REP embedded scripts executed as a main script that control general functionality and call child scripts respect to the tree hierarchy while child script handle certain part of them. Additionally, V-REP supports add-on and plug-ins that can be used to extend the functionality of the existing models or objects. V-REP remote API clients embedded as a virtual code prototype used for call function remote. Besides, V-REP computes the dynamic task and perform the kinematic simulation.

IV. Comparison of Simulation Software

We present a summary of simulation software information in Table I. The features of the simulation software are classified according to the following criteria:

- Developed by: states who/what organization develop the simulation software.
- License type: describes the licensing scheme whether "GPL" license that is open-source software or/and "Proprietary" license that is commercial software.
- Supported operating system (OS): states the operating system supported for software installation.
- Programming language: describe the supported programming language by the simulation software. Some software has it own programming language.
- Visualization type: what kind of the models and user interface visualization offer by the simulation software. It can be in "2D", "2.5D" and/or "3D".

As we can see in Table I, NetLogo, GAMA, and Player/Stage operate with GPL license. These features give courage to the user to explore the software easily without purchasing the software. In contrast, Webots operates with a proprietary license that makes it an inconvenience to explore. Meanwhile, V-REP comes with both free and commercial package. Users can choose either the free pro educational version or commercial pro version. Some software provides model libraries and some embedded with pre-built simulation. In V-REP, the user can just drag and drop the robot model from libraries. Although NetLogo and GAMA are designed for agent-based simulation, the user can assume that the agents represent robot models.

Table II shows a summary of the advantages and disadvantages of each simulation tools. Even powerful tools have its downside. Based on the table, readers can go through which simulation software is suitable for them to perform multi-robots experiments. We can summarize that each simulation software has their special features that match user ability.

To select the best tools for the non-expert, the details criteria of the software should be considered. The best tools for non-expert must ease the user in terms of the installation process, programming difficulty, models or libraries provided, and availability of documentation and references. The tools should also be able to run algorithms and possess the ability to transfer codes to real robots. For the installation process, if the software installation takes only a few minutes we assumed it as "easy". Whereas "difficult" is when the user needs to install other plug-ins or software beforehand and require certain setup to complete the installation. Programming difficulty describes the ease of learning the programming language. It can be "easy" if the user is familiar with the language, or the language is simple and require a short time to understand. On the other hand, it is "difficult" if the software comes with their complex programming language and if common programming language takes a longer time to understand. Some software provides model libraries for user ease. Software that "provided" model libraries is better than the one that "minimally provided" libraries because the existing models is very helpful especially for the non-expert user. Any simulation software "able" to run algorithms, but the simple software features or interface is "preferable" to test the algorithm. The documentation and references should be "extensive" enough to help the user. The ability to transfer to a real robot can be categorized in "yes" or "no". A "yes" means the programming interface can be ported or transfer the results to real robot otherwise is "no".

Referring to Table III that we come up with, the simulation software is listed in a sequence based on several factors/cases. All the preferences should follow the sequence. If the user wants to build a simulation quickly but have less expertise in computer programming, they can choose according to this preferences; NetLogo, GAMA, Webots, Player/Stage, and V-REP. If the user wants to apply and transfer the simulation to a real robot, they can go only with this preferences; Player/Stage, V-REP, and Webots. If the user just wants to run the algorithm and not require robots model, it is advisable to choose either NetLogo or GAMA.

TABLE I. SIMULATION SOFTWARE FEATURES

Simulation Software	Features				
	Developed by	License Type	Supported OS	Programming Language	Visualization Type
NetLogo	Northwestern University, Center for Connected Learning (CCL) and Computer-Based Modeling	GNU GPL	Cross-platform (any Java Virtual Machine)	NetLogo	2D, 3D
GAMA	UMMISCO IRD / UPMC International Research Unit	GNU GPL	Windows, Linux, Mac OS X	GAMA Modeling Language (GAML)	2D, 3D
Webots	Cyberbotics Ltd	Proprietary	Windows, Linux, Mac OS X	C / C++, Java, Matlab, Python	3D
Player/Stage	Brian Gerkey, Richard Vaughan, Andrew Howard, Nathan Koenig	GNU GPL	Linux, Solaris and Mac OSX (Darwin)	Player: any; Stage: C / C++, Python, Java	2.5D
V-REP	Coppelia Robotics	Proprietary, GNU GPL	Cross-platform	Lua, C / C++, Java, Python, Matlab, Urbi, Octave	3D

TABLE II. ADVANTAGES AND DISADVANTAGES OF EACH SOFTWARE

Simulation Software	Advantages	Disadvantages
NetLogo	Provide complete documentation	Allow for 3D but limited, not suitable for the advanced programmer
GAMA	Active mailing list, allow for mapping	Complex environment, rarely used in the robotics field
Webots	Allow to transfer to real robots	Commercial license
Player/Stage	Allow to transfer to real robots	High learning curve, no main user interface
V-REP	Offers extensive API	Difficult for novice user

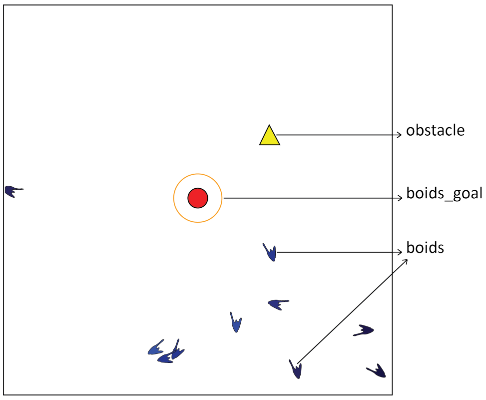

Fig. 1. Initial state of the Boids simulation environment.

V. CASE STUDY

In this section, we presented a simple demo in GAMA platform to guide the non-expert to perform a simple model modification. Noted that GAMA platform is not a robotics software thus the agents itself instantiate the robots. We could run the simulation on any simulation software. Instead, we take this chance to explore GAMA and to perform the demonstration since GAMA is relatively new and rarely used in the robotics area. It is a convenience for the non-expert to learn GAML by a simple demonstration.

We used GAMA platform (version 1.6.1) and a Windows-based netbook for our simulation. The installation of GAMA requires around 200MB available disk space and 4GB RAM (recommended). The requirement of GAMA installation is Java 1.6, and the latest JDK is preferable.

We used boids model in GAMA library to show the demo. As user follow the steps precisely, they will end up the model modification successfully. Firstly, open the existing Boids model in directory Models library > Toy Models > Boids > models. The simulation shows that the birds (`boids`) are wandering around and follow the circle target (`boids_goal`).

The background should be clean as we tried to remove the background image. However, the environment still crowded because it is filled with too many boids. Hence, we cut down the number of boids and number of obstacles by reducing from 100 to 10 and from 5 to 1 in line 3 and line 4 respectively. We want the initial position of `boids_goal` is at the center of the environment as shown in Fig. 1. Since the location of the `boids_goal` is pointed to the `goal`, the modification should be made at the `goal` code on line 22. To get a uniform movement of `boids_goal`, we want to change its behavior. Originally, the `boids_goal` is moving randomly. We want the `boids_goal` moving circularly in a counterclockwise direction. Follow codes in line 51 until line 54 in Fig. 2 to replace wander behavior to move behavior. The variable a here represent the radius of circular movement of the moving `boids_goal`. Fig. 2 shows the lines of code after modification is done.

To demonstrate the usability of the GAMA platform, four changes were made to execute a simple demo. The modification involved are background image removal, reducing the number of agents, the initial location of agent and the movement of the agents. Fig. 3 show the sample sequences of the simulation. Fig. 3a shows the `boids` started to change the direction and follow the `boids_goal`. The moving boids then diverge to avoid the obstacle as in Fig. 3b and converge as soon as they pass the obstacle. Fig. 3c shows the `boids` still wandering around without exceeding the desired range. Fig. 3d shows after around 355 cycles, the `boids_goal` almost complete a round of the circular path. As the simulation keeps running, the `boids_goal` maintains its circular path in the same counter-clockwise direction.

VI. CONCLUSION

In conclusion, the multi-robot experiment can be run either on robotics simulation or agent-based simulation platform. However, the selection of the right simulation software is vital to producing a smooth experiment along the way. Each simulation software has its advantages and disadvantages.

TABLE III. COMPARISON BETWEEN EACH SIMULATION SOFTWARE

Simulation Software	Criteria					
	Installation Process	Programming Difficulty	Models/Libraries Provided	Run Algorithms	Availability of References	Transfer to Real Robots
NetLogo	Easy	Easy	Provided	Preferable	Extensive	No
GAMA	Easy	Easy	Provided	Preferable	Not extensive	No
Webots	Difficult	Easy	Provided	Able	Extensive	Yes
Player/Stage	Difficult	Difficult	Minimum	Able	Not extensive	Yes
V-REP	Easy	Difficult	Provided	Able	Extensive	Yes

```
model boids
global torus: torus_environment{
    int number_of_agents <- 10 min: 1 max: 1000000;
    int number_of_obstacles <- 1 min: 0;
                        • • •
    point goal <- {(width_and_height_of_environment / 2) + 1, (width_and_height_of_environment /2) + 1 };
    list images of: file <- [file('../images/bird1.png'),file('../images/bird2.png'),file('../images/bird3.png')];
    int xmin <- bounds depends_on: [bounds];
    int ymin <- bounds depends_on: [bounds];
    int xmax <- (width_and_height_of_environment - bounds);
    int ymax <- (width_and_height_of_environment - bounds);
    int a <- 2;
                        • • •
entities {
    species name: boids_goal skills: [moving] {
        const range type: float <- 20.0;
        const size type: float <- 10.0;

        reflex move {
            location <- { location.x + (a * cos (time) ), location.y + ( (-a) * sin (time) ) };
            goal <- location;
        }
                        • • •
    output {
        display Sky refresh_every: 1 {
            species boids aspect: image;
            image 'background' file:'../images/sky.jpg';
            species boids_goal;
            species obstacle;
        }
    }
```

Fig. 2. The boids model codes after modifications. The boxes are the parts where the modifications were done.

From the research, NetLogo seems like a right software for a non-expert to handle. GAMA platform is recommended for building a multi-agent experiment. Webots and V-REP provide robot models that ease the users to just apply the algorithm to the simulation. Meanwhile, Player/Stage is also an excellent tool but the matter is the steep learning curve. The comparison of the features of all five simulation software is discussed in Section IV. Then, the selection of the best tools is made based on six criteria such as the installation process, programming difficulty, models or libraries provided, ability to run algorithms, availability of references and ability to transfer to real robots. This paper could assist non-experts to make a decision on the simulation software for their use cases or situation.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the funding provided by the Malaysia Ministry of Education (Grant No.: ERGS/2013/FTMK/ICT02/UTEM/03/01/E00028) and the support given by University Teknikal Malaysia Melaka. We also appreciate the facilities provided by Creative Media Lab, Faculty of Information and Communication Technology and the Center of Excellence for Robotics and Industrial Automation.

REFERENCES

[1] A. Staranowicz and G. L. Mariottini, "A Survey and Comparison of Commercial and Open-source Robotic Simulator Software," in *Proceedings of the 4th International Conference on PErvasive Technologies Related to Assistive Environments*, ser. PETRA '11. New York, NY, USA: ACM, 2011, pp. 56:1–56:8.

[2] S. F. Railsback, S. L. Lytinen, and S. K. Jackson, "Agent-based Simulation Platforms: Review and Development Recommendations," *Simulation*, vol. 82, no. 9, pp. 609–623, 2006.

[3] A. Drogoul, E. Amouroux, P. Caillou, and B. Gaudou, "GAMA: Multi-Level and Complex Environment for Agent-Based Models and Simulations," in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems*, 2013, pp. 1361–1362.

[4] The Player Project. (2015, Jul.) Player Project. [Online]. Available: http://playerstage.sourceforge.net

[5] Coppelia Robotics. (2015, Jul.) v-rep virtual robot experimentation platform. [Online]. Available: http://www.coppeliarobotics.com

[6] D. A. Robertson, "Review of 'Agent-Based Modeling Toolkits'," *Academy of Management Learning Education*, vol. 4, no. 4, pp. 525–527, 2005.

[7] A. Drogoul. (2015, Jul.) gama-platform - Agent-based, spatially explicit, modeling and simulation platform. UMMISCO IRD/UPMC International Research Unit. [Online]. Available: https://code.google.com/p/gama-platform

[8] S. Tisue and U. Wilensky, "Netlogo: A Simple Environment for Modeling Complexity," in *International Conference on Complex Systems*. Boston, MA, 2004, pp. 16–21.

[9] E. G. Macatulad and A. C. Blanco, "3DGIS-Based Multi-Agent Geosimulation and Visualization of Building Evacuation Using GAMA Platform," *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XL-2, pp. 87–91, 2014.

[10] U. Wilensky. (2015, Jul.) NetLogo. [Online]. Available: http://ccl.northwestern.edu/netlogo/

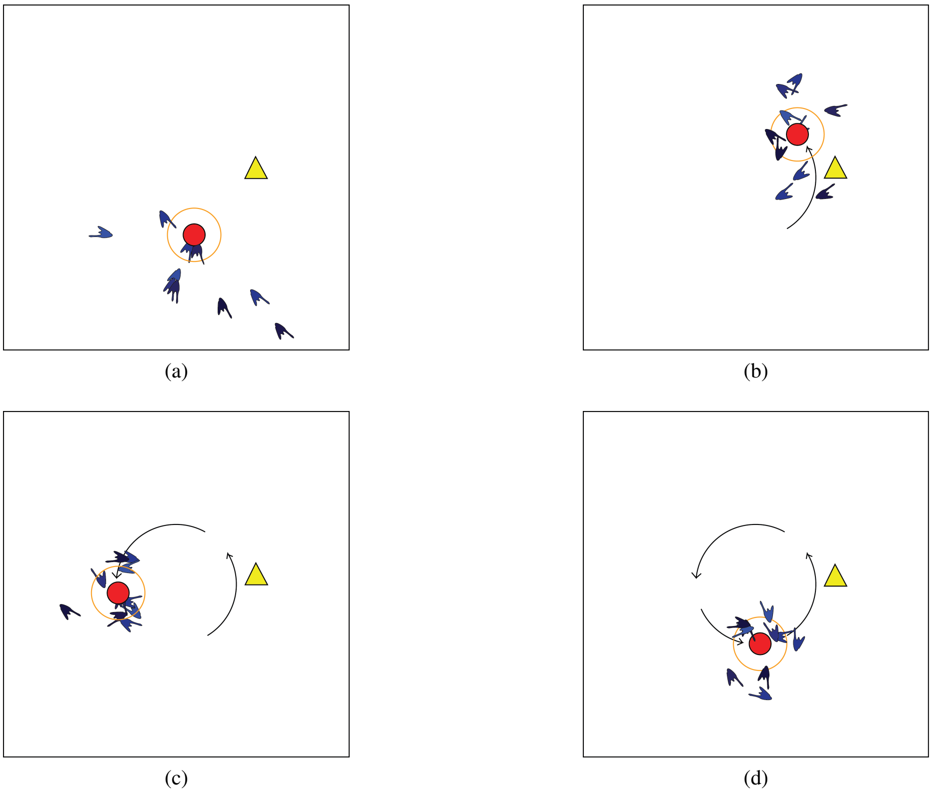

Fig. 3. Sample sequence of boids simulation in GAMA platform; (a) shows the boids started to change the direction; (b) moving boids then diverge to avoid the obstacle; (c) boids converge as soon as they pass the obstacle; (d) boids still wandering around without exceeding the desired range.

[11] S. Tisue and U. Wilensky, "NetLogo: Design and Implementation of a Multi-Agent Modeling Environment," in *Proceedings of the Agent Conference*, 2004, pp. 7–9.

[12] NetLogo. (2015, Jul.) NetLogo 5.2 User Manual: Tutorial #1: Models. [Online]. Available: http://ccl.northwestern.edu/netlogo/docs/tutorial1.html

[13] A. Drogoul. (2015, Jul.) GAMA GitHub. [Online]. Available: https://github.com/gama-platform

[14] u.-A. Võ, A. Drogoul, and J.-D. Zucker, "Multi-Level Agent-Based Modeling: a Generic Approach and an Implementation," in *Advanced Methods and Technologies for Agent and Multi-Agent Systems*, ser. Frontiers in Artificial Intelligence and Applications. IOS Press, 2013, vol. 252, pp. 91–101.

[15] Cyberbotics Ltd. (2015, Jul.) Webots: robot simulation software. [Online]. Available: https://www.cyberbotics.com

[16] M. S. Couceiro, P. A. Vargas, and R. P. Rocha, "Bridging the Reality Gap Between the Webots Simulator and e-puck Robots," *Robotics and Autonomous Systems*, vol. 62, no. 10, pp. 1549–1567, 2014.

[17] N. M. H. Basri, K. S. M. Sahari, S. S. K. Mohideen, N. S. Roslin, and A. Anuar, "Development of 3D Boiler Header Template using Webots," *Procedia Engineering*, vol. 41, pp. 1463–1468, 2012.

[18] L. Hohl, R. Tellez, O. Michel, and A. J. Ijspeert, "Aibo and Webots: Simulation, Wireless Remote Control and Controller Transfer," *Robotics and Autonomous Systems*, vol. 54, no. 6, pp. 472–485, 2006.

[19] O. Michel, "Cyberbotics Ltd. webots: Professional Mobile Robot Simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, pp. 39–42, 2004.

[20] B. P. Gerkey, R. T. Vaughan, and A. Howard, "The Player / Stage Project : Tools for Multi-Robot and Distributed Sensor Systems," in *Proceedings of the 2003 International Conference on Advanced Robotics (ICAR 2003)*, 2003, pp. 317–323.

[21] R. Vaughan, "Massively Multi-Robot Simulation in Stage," *Swarm Intelligence*, vol. 2, no. 2-4, pp. 189–208, 2008.

[22] Open Source Robotics Foundation. (2015, Jul.) Gazebo. Open Source Robotics Foundation (OSRF). [Online]. Available: http://gazebosim.org

[23] J. J. O. Barros, V. M. F. D. Santos, and F. M. T. P. D. Silva, "Bimanual Haptics for Humanoid Robot Teleoperation Using ROS and V-REP," in *2015 IEEE International Conference on Autonomous Robot Systems and Competitions*, 2015, pp. 174–179.

[24] E. Rohmer, S. P. N. Singh, and M. Freese, "V-REP: A Versatile and Scalable Robot Simulation Framework," in *IEEE International Conference on Intelligent Robots and Systems*, 2013, pp. 1321–1326.