Smelling on the Edge: Using Fuzzy Logic in Edge Computing to Control an Olfactory Display in a Video Game

Miguel Angel Garcia-Ruiz
School of Computer Science and Technology
Algoma University
Sault Ste. Marie, Canada
miguel.garcia@algomau.ca

Pedro C. Santana-Mancilla
School of Telematics
Universidad de Colima
Colima, Mexico
psantana@ucol.mx

Laura S. Gaytan-Lugo
School of Mechanical and Electrical Engineering
Universidad de Colima
Coquimatlan, Mexico
laura@ucol

Raul Teodoro Aquino-Santos
School of Telematics
Universidad de Colima
Colima, Mexico
aquinor@ucol.mx

***Abstract*—This paper presents a 3D video game project that incorporates smell generated by an olfactory display (an ultrasonic humidifier and a PC fan) controlled by fuzzy logic. In order to improve the olfactory display efficiency, we apply Edge computing by running the fuzzy logic control software on the microcontroller itself and not on the video game computer or a network server. Our video game activates the olfactory display by sending a wireless signal using MQTT data communication protocol to the microcontroller board connected to a local wireless network. The video game objective is to find a virtual lemon in less than 15 seconds, hidden behind many virtual crates. The olfactory display generates a lemon smell when the player is close to the virtual lemon. The fuzzy logic controls the fan speed according to the distance between the virtual lemon and the player's main game view. An early test showed that the fuzzy logic and the MQTT protocol ran efficiently on the microcontroller board. This demonstrates that Edge computing can be useful in simple olfactory display applications.**

Keywords— fuzzy logic, edge computing, olfactory display, smell, odor, human-computer interface, video game, 3D

I. Introduction

We present a work in progress on a game development project that incorporates an olfactory display to support player experience. An olfactory display is a human-computer device that generates and diffuses one or more smells to a user with a purpose [1]. Barfield and Danas also define an olfactory display as "a collection of hardware, software, and chemicals that can be used to represent olfactory information to the virtual environment participant" [2]. Olfactory displays have been proposed and researched with applications in video games, where smell was intended to support player's emotional impact [3], memorability [4] and multisensory gaming experiences [5], among other applications.

We have developed an olfactory display for our game development project controlled by fuzzy logic running on a microcontroller board. The olfactory display takes advantage of its Edge computing capabilities of running the software on the microcontroller efficiently. Fuzzy logic is a soft computing set of techniques applied in artificial intelligence (AI), modeling human logical reasoning with imprecise or vague statements [6]. Fuzzy logic is based on fuzzy sets where "degrees of truth" are assigned to those sets [7]. Edge computing is a distributed computing paradigm bringing data storage and computation closer to the application location, enabling data acquisition and processing closer to the edge of the computer network [8], [9], allowing for small-scale and local data processing [10]. Edge computing can run on a microcontroller with wireless network connection. This is useful to control our olfactory interface, since its video game data processing, the fuzzy logic code, and olfactory interface's actuator control are performed on the microcontroller itself and not on the video game computer or a network server, providing low latency between the microcontroller and the video game. Our Edge computing code was implemented with the Mosquitto library [11], a server/client implementation of the standard Message Queuing Telemetry Transport (MQTT) data communication protocol, used in IoT and Edge computing applications [12]. MQTT uses a publish/subscribe model, generally running with low network overhead [13]. This makes it ideal for running on microcontrollers and Edge computing applications that require fast data communication among devices or sensors using a wireless local area network (WLAN). The MQTT protocol requires a broker for managing the messages among clients [14]. We ran the MQTT broker software on a Raspberry Pi 4 card-size computer. The MQTT clients are an Arduino Nano 33 IoT microcontroller board that controls a humidifier (used for generating a smell) and a PC fan, and a MacBook Pro laptop running a minigame. A solid-state relay connected to the microcontroller board activated the humidifier. The microcontroller board, the laptop and the Raspberry Pi were connected to the same WLAN. The minigame activated the fan and the humidifier by sending messages to the MQTT broker, which were received by the microcontroller board (see Fig. 1). We programmed the client software for the microcontroller board using the ArduinoMqttClient Library for Arduino [15]. The Arduino Nano 33 IoT microcontroller board was powered with a USB power bank, making it a stand-alone application.

In this project, we wanted to see if the olfactory display's response time for generating a smell worked at the right time on Edge computing when the video game is played, validating what was proposed by [16], where Edge computing can improve the response time of AI techniques running on a local computer or microcontroller board, running closer to the video game player in gaming applications.

A. Related Work

The application of smell in entertainment technology has long been suggested and researched as a potential sensory stimulus supporting immersion, including virtual reality, as initially proposed by [2]. Smell has been researched as a way to support multisensory player experience in video games. [17] developed a multisensory virtual reality video game and an olfactory display composed of an Arduino microcontroller board, a small fan and a smoke generator that vaporized "ethereal oil" scents. Their research found that dense smell stimuli should be applied in video games that incorporate smell. In addition, their olfactory display demonstrated that rapid prototyping made with Arduino microcontroller boards is an effective way to develop a simple yet effective smell interface. [18] reviewed early commercial video games that included smell cues using a number of olfactory technologies, including scratch-and-sniff cards. [19] conducted a perception pilot experiment with an olfactory display producing a mixture of coffee and tea smells. 10 participants estimated the smell intensity from different samples, finding that healthy users can identify binary smell mixtures with some accuracy. In another experiment, the same researchers ran an event-based memorization experiment with 15 participants playing a 2D memory video game, where players memorized smell and card locations. Results suggested that previous smell training helped to identify the card locations. The researchers also developed a 3D game where they will test out spatial smell memorizations. On the other hand, in [20], participants were presented with a virtual reality environment where most of them successfully identified the spatial position of two smells, lemon and cedar, generated by a handheld olfactory display housing a microcontroller board.

Fuzzy logic was researched by [21] to control scented dispensers used in a large store as a way to support consumer's emotional state and olfactory experiences. This also proved a successful marketing application. [22] developed an 3D educational game that incorporates a smell working as an olfactory reward for children players that complete a math problem in the game. A smell of pineapple was generated and diffused when the player correctly counted the number of pineapples shown in the game. The amount of smell was generated according to the distance between the player and the game, measuring it using an ultrasonic sensor and fuzzy logic.

II. Our Olfactory Display

We developed an olfactory display consisting of a Noctua NF-P12 PC fan and an off-the-shelf Honeywell Mini Mist™ ultrasonic humidifier containing an essential oil reservoir. We placed 20 drops of Homedics.™ Ellia 100% pure lemon (Citrus Limon) essential oil in the reservoir. Both the fan and the humidifier were controlled by an Arduino Nano 33 IoT microcontroller board [11]. This board model has been used in Internet of Things (IoT) projects (e.g. [12]) and Edge computing applications, for example in [13], taking advantage of its WIFI network connectivity. The humidifier generated the smell and the PC fan directed it towards the player. We used fuzzy logic for controlling the fan speed to regulate the smell intensity, depending on the distance between the virtual lemon and the camera in the game. The closer the player to the virtual lemon is, the faster the fan spins and thus the more lemon smell reaches the player. After various software testing iterations, we found that the smell generation should happen between 0 and 10 units in our 3D game. The complete olfactory display prototype is shown in Fig. 1.

Fig. 1. Our olfactory display prototype.

As Fig. 1 shows, the PC fan is placed on top of the humidifier to diffuse the lemon smell.

III. Fuzzy Logic Implementation

We used the eFLL library for Arduino microcontroller boards [23] for developing our fuzzy logic control software, running on the Arduino Nano 33 IoT board. Our fuzzy rules are as follows:

- IF distance between game camera and virtual lemon = close THEN activate humidifier AND fan speed = high
- IF distance between game camera and virtual lemon = near THEN activate humidifier AND fan speed = medium
- IF distance between game camera and virtual lemon = far THEN activate humidifier AND fan speed = slow

Fig. 2 shows the input fuzzy sets containing the degrees of truth for the distance between the player's view (the game camera position, controlled by the player) and the virtual lemon. The fuzzy sets' values were determined in a number of iterative tests done with the olfactory display and the video game.

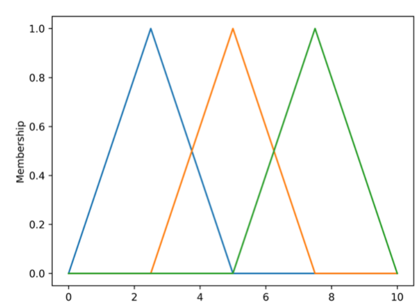

Fig. 2. The input fuzzy sets used in the olfactory display.

The universe of discourse from the three input fuzzy sets shown in Fig. 2 are the input values of [0,10]. A value that belongs to a fuzzy set can have a membership (degree of truth) between 0.0 (the value does not belong to the input fuzzy set) to 1.0 (the value fully belongs to the input fuzzy set), or a membership value in between. For example, as Fig. 2 shows, an input value of 3 is a partial member of the Close fuzzy set with a membership value of 0.8, and it is also a partial member of the Near fuzzy set with a membership value of 0.2. The video game does not activate the olfactory interface when the distance between the virtual lemon and the player exceeds 10 units in the game. This means that the player is very far away from the virtual lemon, thus no smell is generated. As Fig. 2 shows, the three types of input sets are named with three fuzzy logic linguistic variables that define the distance between the virtual lemon and the camera: Close, Near and Far, respectively.

Our fuzzy logic code generated crisp output values used for activating the humidifier and for changing the fan speed for dispersing the smell towards the player. The values ranged from 0 (the fan stops) and 255 (the fan runs at full speed). These values were used because of the 8-bit pulse-width modulation (PWM) output function used by the Arduino microcontroller board. PWM is a technique that is used to control the fan speed by changing the pulse width (square waves) generated by the Arduino microcontroller board [24]. The fuzzy output sets are named with three fuzzy logic linguistic variables that define the fan speed: Slow=[0,64,128], Medium=[64,128,192] and Fast=[128,192,255], respectively.

IV. Our 3D Minigame

We developed a first-person 3D minigame as a proof of concept for testing our olfactory display and fuzzy logic code. The video game was developed using Godot [25], a popular video game engine. The game objective is to find a virtual lemon hidden behind many crates in less than 15 seconds, using the keyboard arrow keys to navigate in the 3D world. The game procedurally generated 200 crates placed randomly, obstructing the virtual lemon view and making it somewhat difficult to find it. The game (shown in Fig. 1) displays a counter counting down from 15 to 0 seconds. If the player is close, near or far to the virtual lemon the olfactory display generates the scented mist with the lemon smell. A Godot script was continuously measuring the distance (in units) between the player and the virtual lemon through their 3D positions (coordinates). When the counter reaches 0, the game stops and the player can no longer use the arrow keys to navigate in the 3D world, that is, moving the game camera around. The olfactory display, its software and the video game used in the project were developed following the Evolutionary Prototyping method [26], taking advantage of its iterative feature. We informally tested our olfactory display and our video game. The test's main task was to find the virtual lemon in our game as fast as possible in less than 15 seconds.

V. Preliminary Results and Conclusions

Our ongoing project demonstrates the use of a simple and effective olfactory display system based on a wireless-capable microcontroller board in a video game. Similar systems can be applied in other domains that may incorporate olfactory stimuli with wireless olfactory display control, for example, in marketing applications. We found in the test that the fuzzy logic code efficiently controlled the PC fan speed. Our test results show that Edge computing can be useful in simple olfactory display applications that generate one smell. There was no perceived delay in managing the distance data between the player and the virtual lemon with MQTT. The MQTT broker running on the Raspberry Pi computer was very stable, running nonstop during the prototype development for various weeks, being an efficient and lightweight option for running our Edge computing application. However, the humidifier took on average 2.5 seconds to start generating its scented mist. Although this delay was tolerable for our video game prototype, it may not be acceptable for fast-paced video games such as shooting games. Our olfactory display system faces some technical limitations. It slowly generates just one smell, limiting its use to very simple olfactory applications in video games. Wirelessly connecting more clients (e.g. various olfactory displays or computers) to the MQTT broker may slow down the data transmission, but further tests are needed to confirm this. In addition, the system does not filter out or eliminate the smell from the environment after it was used in the game. This is a common problem with olfactory displays, e.g. [27], that may be tackled by generating the right amount of smell and using mechanical air filters. Further work will include developing our own ultrasonic mist generator for our olfactory display, in an attempt to generate the mist faster. We will test the usability and player experience of our olfactory display+video game following research ethics protocols.

References

[1] A. Manes, "Web Services Basics," in *Web Services: A Manager's Guide*, 1st Edition., Addison-Wesley Professional, 2003, pp. 27–46.

[2] W. Barfield and E. Danas, "Comments on the use of olfactory displays for virtual environments," *Presence: Teleoper. Virtual Environ.*, vol. 5, no. 1, Art. no. 1, Jan. 1996, doi: 10.1162/pres.1996.5.1.109.

[3] C. Spence, "Scenting Entertainment: Virtual Reality Storytelling, Theme Park Rides, Gambling, and Video-Gaming," *i-Perception*, vol. 12, no. 4, Art. no. 4, Jul. 2021, doi: 10.1177/20416695211034538.

[4] M. A. Garcia-Ruiz, A. Edwards, R. Aquino-Santos, O. Alvarez-Cardenas, and M. G. Mayoral-Baldivia, "Integrating the Sense of Smell in Virtual Reality for Second Language Learning," Nov. 2008, pp. 2647–2652. Accessed: Mar. 31, 2022. [Online]. Available: https://www.learntechlib.org/primary/p/30042/

[5] T. Nakamoto, S. Otaguro, M. Kinoshita, M. Nagahama, K. Ohinishi, and T. Ishida, "Cooking Up an Interactive Olfactory Game Display," *IEEE Computer Graphics and Applications*, vol. 28, no. 1, Art. no. 1, Jan. 2008, doi: 10.1109/MCG.2008.3.

[6] P. Cintula, C. G. Fermüller, and C. Noguera, "Fuzzy Logic," in *The Stanford Encyclopedia of Philosophy*, Winter 2021., E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2021. Accessed: Apr. 11, 2022. [Online]. Available: https://plato.stanford.edu/archives/win2021/entries/logic-fuzzy/

[7] L. A. Zadeh and R. A. Aliev, *Fuzzy Logic Theory And Applications: Part I And Part Ii*. World Scientific Publishing, 2018.

[8] W. Z. Khan, E. Ahmed, S. Hakak, I. Yaqoob, and A. Ahmed, "Edge computing: A survey," *Future Generation Computer Systems*, vol. 97, pp. 219–235, Aug. 2019, doi: 10.1016/j.future.2019.02.050.

[9] P. C. Santana-Mancilla, L. E. Anido-Rifón, J. Contreras-Castillo, and R. Buenrostro-Mariscal, "Heuristic Evaluation of an IoMT System for Remote Health Monitoring in Senior Care," *IJERPH*, vol. 17, no. 5, p. 1586, Mar. 2020, doi: 10.3390/ijerph17051586.

[10] K. Cao, Y. Liu, G. Meng, and Q. Sun, "An Overview on Edge Computing Research," *IEEE Access*, vol. 8, pp. 85714–85728, 2020, doi: 10.1109/ACCESS.2020.2991734.

[11] R. A. Light, “Mosquitto: server and client implementation of the MQTT protocol,” *JOSS*, vol. 2, no. 13, Art. no. 13, May 2017, doi: 10.21105/joss.00265.

[12] “MQTT - The Standard for IoT Messaging,” May 05, 2022. https://mqtt.org/ (accessed May 05, 2022).

[13] H. Koziolek, S. Grüner, and J. Rückert, “A Comparison of MQTT Brokers for Distributed IoT Edge Computing,” in *Software Architecture*, vol. 12292, A. Jansen, I. Malavolta, H. Muccini, I. Ozkaya, and O. Zimmermann, Eds. Cham: Springer International Publishing, 2020, pp. 352–368. doi: 10.1007/978-3-030-58923-3_23.

[14] K. Sahlmann and T. Schwotzer, “Ontology-based virtual IoT devices for edge computing,” in *Proceedings of the 8th International Conference on the Internet of Things*, Santa Barbara California USA, Oct. 2018, pp. 1–7. doi: 10.1145/3277593.3277597.

[15] *ArduinoMqttClient Library for Arduino*. Arduino Libraries, 2022. Accessed: May 10, 2022. [Online]. Available: https://github.com/arduino-libraries/ArduinoMqttClient

[16] A. Pardos, A. Menychtas, and I. Maglogiannis, “On unifying deep learning and edge computing for human motion analysis in exergames development,” *Neural Comput & Applic*, vol. 34, no. 2, Art. no. 2, Jan. 2022, doi: 10.1007/s00521-021-06181-6.

[17] E. Kruijff, A. Marquardt, C. Trepkowski, J. Schild, and A. Hinkenjann, “Enhancing User Engagement in Immersive Games through Multisensory Cues,” in *2015 7th International Conference on Games and Virtual Worlds for Serious Applications (VS-Games)*, Skövde, Sweden, Sep. 2015, pp. 1–8. doi: 10.1109/VS-GAMES.2015.7295773.

[18] S. Niedenthal, “Skin Games: Fragrant Play, Scented Media and the Stench of Digital Games,” *EJCGC*, vol. 6, no. 1, Art. no. 1, May 2012, doi: 10.7557/23.6141.

[19] J. K. Olofsson, S. Niedenthal, M. Ehrndal, M. Zakrzewska, A. Wartel, and M. Larsson, “Beyond Smell-O-Vision: Possibilities for Smell-Based Digital Media,” *Simulation & Gaming*, vol. 48, no. 4, Art. no. 4, Aug. 2017, doi: 10.1177/1046878117702184.

[20] S. Niedenthal, P. Lundén, M. Ehrndal, and J. K. Olofsson, “A Handheld Olfactory Display For Smell-Enabled VR Games,” in *2019 IEEE International Symposium on Olfaction and Electronic Nose (ISOEN)*, May 2019, pp. 1–4. doi: 10.1109/ISOEN.2019.8823162.

[21] C. Vuppalapati, R. Vuppalapati, S. Kedari, A. Ilapakurti, J. S. Vuppalapati, and S. Kedari, “Fuzzy Logic Infused Intelligent Scent Dispenser For Creating Memorable Customer Experience of Long-Tail Connected Venues,” in *2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, Jul. 2018, vol. 1, pp. 149–154. doi: 10.1109/ICMLC.2018.8527046.

[22] M. Garcia-Ruiz, B. Kapralos, and G. Rebolledo-Mendez, “Towards Effective Odor Diffusion with Fuzzy Logic in an Olfactory Interface for a Serious Game,” in *HCI International 2021 - Late Breaking Papers: Multimodality, eXtended Reality, and Artificial Intelligence*, Cham, 2021, pp. 3–16. doi: 10.1007/978-3-030-90963-5_1.

[23] A. J. Alves, *eFLL*. 2022. Accessed: Mar. 27, 2022. [Online]. Available: https://github.com/alvesoaj/eFLL

[24] “Basics of PWM (Pulse Width Modulation) | Arduino Documentation,” Apr. 26, 2022. https://docs.arduino.cc/learn/microcontrollers/analog-output (accessed Apr. 26, 2022).

[25] “Godot Engine - Free and open source 2D and 3D game engine,” Mar. 29, 2022. https://godotengine.org/ (accessed Mar. 28, 2022).

[26] S. McConnell, *Rapid development: taming wild software schedules*. Redmond, Wash: Microsoft Press, 1996.

[27] G. Tsaramirsis, M. Papoutsidakis, M. Derbali, F. Q. Khan, and F. Michailidis, “Towards Smart Gaming Olfactory Displays,” *Sensors*, vol. 20, no. 4, Art. no. 4, Jan. 2020, doi: 10.3390/s20041002.