A Fuzzy Logic Based System for Cloud-based Building Information Modelling Rendering Optimization in Augmented Reality

Hugo Leon-Garza and Hani Hagras
Computational Intelligence Group, School of Computer Science and Electronic Engineering
University of Essex
Colchester, UK
hleong@essex.ac.uk

Anasol Peña-Rios, Gilbert Owusu and Anthony Conway
BT Labs
BT Plc
Adastral Park, Martlesham Heath, Ipswich, UK
anasol.penarios@bt.com

***Abstract*— In recent years, Building Information Modelling (BIM) has become the standard for managing the lifecycle of a building. The metadata embedded to BIM models can be used onsite to enhance field worker's view with relevant information about the task. Augmented Reality (AR) is a natural candidate for enabling onsite interactions with BIM models and data, due to its ability for overlaying digital information on top of real-world objects through a digital display. However, the complexity of BIM models and the limited hardware capabilities of AR-ready devices (e.g. head mounted and mobile devices), makes it difficult to provide reliable visualizations, decreasing considerably application's performance. This paper presents a type-1 Fuzzy Logic System (T1FLS) that helps optimizing loading of BIM 3D models in an AR application. Experiments results show that the proposed T1FLS has an average frame per second (FPS) rate of 16.33 in the selected AR headset (Microsoft HoloLens). The FPS rate when using the proposed T1FLS is in average 2.33 times better than using a fixed batch size when loading BIM 3D objects.**

Keywords*— *Augmented Reality, Fuzzy Logic System, Building Information Modelling (BIM).

I. INTRODUCTION

The increasing complexity of maintenance and monitoring related tasks in utility companies has been triggered by diverse factors, such as the mix of legacy and new systems and infrastructure. In many situations, field workers still rely on the use of paper footprints to identify infrastructure components and perform their assigned tasks [1]. By adopting the use of digital representations of facilities and assets, utility companies could move towards a digital documentation. A paperless documentation could bring benefits like reduction of time spent looking for information and efficient administrative processes [2].

The visualization of complex data using digital representations of structures, such as Building Information Modelling (BIM) objects, could help field workers identify patterns and improve decision making when doing provisioning or maintenance tasks [3]. Augmented Reality (AR) could be a key technology for supporting field workers as it can provide enhanced experiences of real-world situations by overlaying key information and three-dimensional (3-D) visualizations when needed. This can reduce operational costs and improve productivity by leveraging and augmenting key work flows and procedures [4], [5]. However, one of the challenges of combining BIM models with an AR system is to identify a hardware platform capable of executing and displaying large amounts of complex information [1]. A BIM model can contain thousands of 3D objects and related metadata, and mobile or AR devices might be limited in performance or memory capabilities.

In this paper we present a process for managing and optimizing the rendering of large amounts of 3D objects into the user's field of view without a reduction in performance; adjusting it to different device capabilities. The main component in the process is a Type-1 Fuzzy Logic System (T1FLS) that determines the maximum amount of 3D objects that can be loaded into scene during each frame. The input parameters that feed the T1FLS are the distance between each 3D object and user's spatial location, the current memory usage level, the current average frames per second (FPS) and the downloading speed.

The paper is organized as follows: section II presents a brief overview on BIM and AR. Section III describes the architecture of the proposed system and the T1FLS implementation. Section IV describes the experiments and results. Finally, in section V we present our conclusions and future work.

II. BRIEF OVERVIEW ON BIM AND AR

BIM is a cooperative process for creating and managing information on a construction project across the building lifecycle [6]. One of the key outputs of this process is the BIM object or model, the digital description of every aspect of the built asset.

A BIM object is a data-rich, object-oriented, intelligent and parametric digital representation of a facility or building, which can be used to make decisions and improve the process of delivering the facility and its maintenance management [7]. BIM models contain large amount of data that can be exploited for tasks related to building and facility management. The elements

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

in the model have properties and relationships, hence they are considered data-rich [8]. All this information can be used to model other dimensions such as time and cost of a project [9].

For example, the work in [5] showed how BIM data could improve decision making in maintenance planning tasks by providing the project stakeholders with a detailed visualization of large amounts of data in a BIM model. Moreover, given its data-rich characteristics, BIM models could be combined with geographic information systems (GIS) data to perform navigation route planning [10]. However, visualization of this complex models has been limited to complex software or 2D visualizations, which adds a layer of complexity for users, as they need to have the right knowledge and expertise to transform this 2D data into real 3D physical environments. Augmented reality (AR) offers a possibility for improving onsite workers' tasks by enabling visualization of these models; giving an overall clarity of the project [11].

Augmented Reality (AR) forms part of Milgram's reality-virtuality continuum [10], which works as a classification that defines different degrees between the real and the virtual world. In this continuum, AR is placed closer towards the real environment end. It allows the user to see the real world, with virtual data superimposed upon or composited with the real world [11]. In other words, AR is a view of the real world with additional information or objects that supplement this reality, rather than completely replacing it. This additional information comes from digital files that can be stored either locally or remotely.

AR has been used before to support field engineering decision-making, for example by allowing visualizations of immersive 3D georeferenced data [12], [13]. According to [14], the ability for displaying digital information on top of users' field of view is what makes AR a perfect partner for BIM. In [15], the authors present different use cases of how BIM + AR can be applied to different building lifecycle activities. However, combining AR with BIM models for onsite tasks is not simple. The challenge involves the need of having portable hardware, which in most of the current implementations cannot be moved to the facility's location [16].

BIM models are complex and size heavy data objects, which can be composed of millions of entities and reach sizes over 3 GB [17]. This problem often limits choices of hardware that can be used for visualizing and interacting with them. This constraint usually leads to the use of high-end laptops to solve any performance issues [18]. However, this solution is not portable due to the hardware being big, heavy, fragile and expensive, which makes it not ideal for outdoor conditions. A solution proposed in [17], was to pre-process BIM models to create lightweight versions able to have a faster rendering. This solution was tested in a web environment with successful results. In this paper, we introduce a different approach for loading complex BIM models for onsite activities in AR enabled devices. The proposed solution and its results are presented in the next sections.

III. The Proposed T1FLS for Cloud-based BIM Rendering Optimization in Augmented Reality

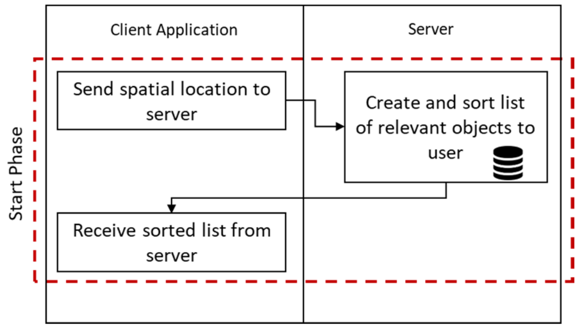
A. *An Overview of the proposed system*

The AR application created for visualizing BIM models for field engineering task support is a cloud-based solution that downloads and displays BIM models as 3D objects at runtime based on user's spatial location. The challenge when loading multiple objects every frame is that usually the application overloads and it becomes unusable. To avoid this kind of overload the number of objects loaded per frame should be adjusted according to the capabilities of the device.

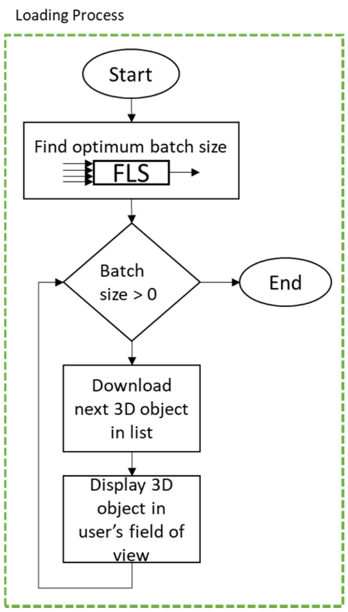
Defining and dynamically adjusting the number of objects to be loaded is a complex task, since not all experts will agree on the limits and the batch size should not suffer big changes for barely exceeding these limits. By using Fuzzy Logic, the proposed system will take expert knowledge and define flexible boundaries to be used in the batch optimization process. Fuzzy logic excels in dealing with the imprecise knowledge associated to a problem [19]–[24]. In fuzzy logic, an input value can belong to different fuzzy sets and its degree of membership is a crisp number in the interval [0,1] [25]–[27]. The association of a crisp value to a fuzzy set makes the system capable of reasoning based on rules with linguistic labels [28]. This removes the need for hard limits and mimics the human expert decision-making process, therefore, a Fuzzy Logic System (FLS) is proposed as a solution for determining the optimum batch size of 3D objects to be loaded each frame. To find the optimum number, the proposed T1FLS implementation uses an initial batch size as the starting point and based on the input evaluation it determines how much the batch should be reduced to achieve an optimal performance in the client application (e.g. an output of 83 from the system will reduce the initial batch size by 83%).

The complete process for loading 3D objects consists of two phases: start and online (Fig. 1). Fig. 1a describes the *start phase*, where the application sends user's spatial location to the remote server and receives the list of objects relevant to user's position. The list is sorted based on the object's priority to be rendered, objects that are closer to the user and have a less complex geometry get a higher priority. Once the *start phase* is completed the application switches to the *online phase*.

Fig. 1b shows the flowchart diagram for the optimized loading process. This process is part of the *online phase* and it is executed at each frame of the application. The process starts by using the T1FLS to determine the batch size that will allow the client application to run at an optimal performance level, this is done by using 4 inputs which are: a) distance between the next object and the user, b) the current memory level usage, c) the average frames per second (FPS) rate and d) the download speed.



(a)



(b)

Figure 1. a) Process diagram for getting the list of relevant objects to be loaded. b) Flow diagram of the loading process that happens during the Online Phase at each frame.

Based on these inputs the T1FLS decides the optimum size of 3D objects batch to avoid an FPS rate drop of the application. After this has been calculated, the application requests the object(s) to the cloud server, downloads them in the local device, and renders them in user's field of view (Fig. 1b). An important consideration is that multiple requests in a single frame will overload the application; therefore, finding the optimum batch size is important for avoiding overload and provide a better user experience.

B. *The Fuzzy Logic System Design*

The proposed T1FLS consists of 4 inputs and 1 output, with different linguistic labels. All the functions were created using expert knowledge and optimized using a heuristic approach. Fig. 2 illustrates membership functions for each input as follows:

a) Average frames per second (FPS) rate, used to understand the application's performance at runtime.

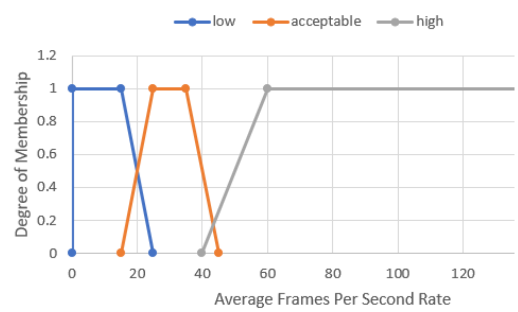
b) Distance between the next object and the user, used to determine the importance of loading the next object(s), objects close to user should be loaded in the least amount of time possible.

c) Current memory level usage, used to understand when to stop loading objects.

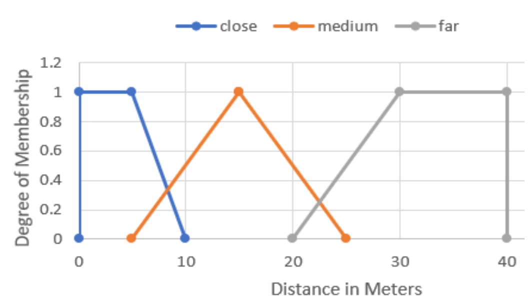
d) Download speed, used to determine how fast can objects be downloaded.

The output of the T1FLS determines the optimal batch size for 3D object rendering (Fig. 2e). Using expert knowledge, we generated a 72 rule set covering all the linguistic variables defined. After that, the rule base was optimized by identifying sets of rules that could be replaced with a rule of one or two antecedents. The simplifications made were as follows: (1) a single antecedent rule with memory level "*very high*" and output "*no batch*" was used to replace all rules that had the same antecedent, (2) a rule with "*low*" FPS as antecedent and "*no batch*" as output was used to replace all rules that contained the same antecedents, (3) a rule with "*far*" distance and "*bad*" download speed antecedents and "*no batch*" as output replaced all the rules that had both of the antecedents, (4) a rule with "*high*" memory level and "*high*" FPS as antecedents and "*small*" batch as output was used to replace all rules that had both of this antecedents. The resulting 29 rule base is shown in Table 1.

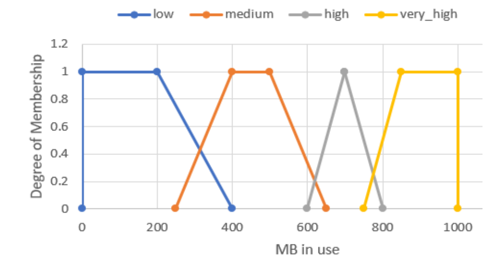
The values for distance, memory level and FPS were read every frame. The download speed input value is computed when the client application starts, and then every 200 frames to reduce the overhead of doing it every frame. The 200 frames translate to 10 – 12 seconds since the application is expected to run at an average performance rate between 18 and 20 frames per second. This interval time was decided based on heuristic methods. The T1FLS uses a singleton fuzzifier, the minimum t-norm to represent the AND logical connector and the center of sets defuzzification [25].



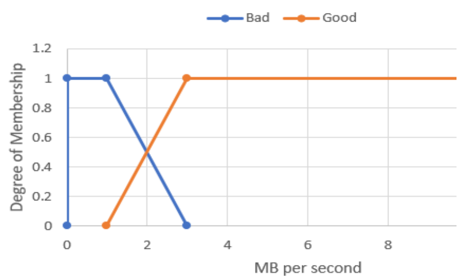
(a)



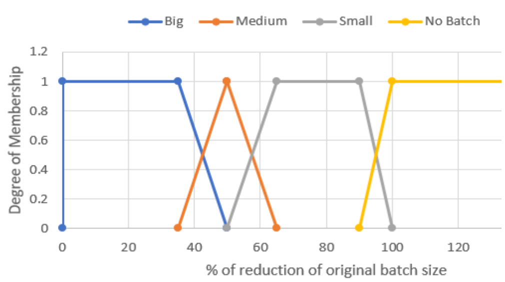
(b)



(c)



(d)



(e)

Figure 2. Membership functions. Input: a) FPS rate level, b) Distance, c) Memory usage, d) Download speed. Output: e) 3D objects' loading batch size.

I. EXPERIMENTS AND RESULTS

The client application was developed using Unity 3D (version 2018.4.0f1). Unity 3D 1 is a cross-platform game engine for creating interactive 3D content. The BIM data used belongs to two residential buildings comprising 379 apartments, on 1.58 acres located in Birmingham, UK.

A pre-processing pipeline was developed to separate the whole BIM model into single files for each object. This pipeline was developed using Python and the IfcOpenShell library. After processing the original BIM model, 126,252 single 3D object files in Wavefront (.obj) format were obtained. These files were stored in a cloud repository.

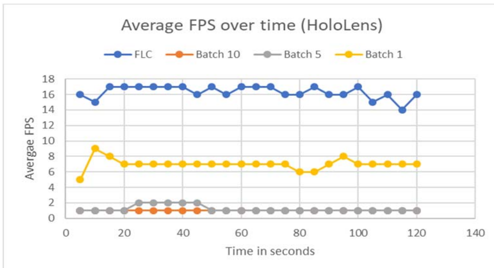
The client application was tested in two different devices: a laptop computer and a head-mounted display. The laptop used was a Lenovo ThinkPad P52 with an Intel Core i7-8850H processor and 24GB of RAM. The head-mounted display used was a Microsoft HoloLens 1 device (shown in Fig. 3). This device has a 32-bit intel architecture processor plus a custom-built Microsoft Holographic processing unit and 2GB RAM, from which only 900MB are available to be used by applications.

TABLE I. OPTIMAL BATCH SIZE FOR 3D OBJECT RENDERING RULE BASE

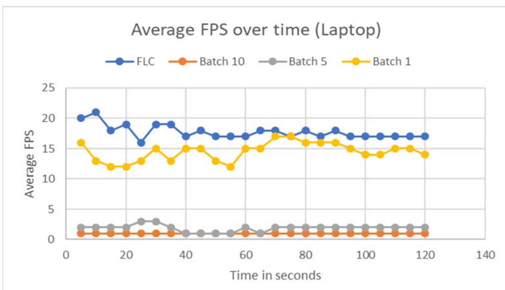
Rule	Antecedents				Consequents
	Distance	***Memory***	***FPS***	***Downloading Speed***	***Batch Size***
R1	-	very high	-	-	no batch
R2	-	high	high	-	small
R3	-	-	low	-	no batch
R4	far	-	-	bad	no batch
R5	close	low	acceptable	good	medium
R6	close	low	acceptable	bad	small
R7	close	low	high	good	big
R8	close	low	high	bad	medium
R9	close	medium	acceptable	good	small
R10	close	medium	acceptable	bad	small
R11	close	medium	high	good	big
R12	close	medium	high	bad	medium
R13	close	high	acceptable	good	small
R14	close	high	acceptable	bad	no batch
R15	medium	low	acceptable	good	medium
R16	medium	low	acceptable	bad	small
R17	medium	low	high	good	big
R18	medium	low	high	bad	medium
R19	medium	medium	acceptable	good	small
R20	medium	medium	acceptable	bad	no batch
R21	medium	medium	high	good	big
R22	medium	medium	high	bad	medium
R23	medium	high	acceptable	good	small
R24	medium	high	acceptable	bad	no batch
R25	far	low	acceptable	good	small
R26	far	low	high	good	medium
R27	far	medium	acceptable	good	small
R28	far	medium	high	good	medium
R29	far	high	acceptable	good	no batch

For AR applications, a quality criteria includes the frames per second (FPS) rate [29]. The average FPS rate describes how many updates are completed per second and is used as a performance measure. If the rate drops to 1 FPS, the application becomes a set of static images. As part of the evaluation for the system proposed, we compared the performance of the T1FLS in the different devices with the use of different fixed batch sizes.

Figure 3. Microsoft HoloLens version 1.



(a)

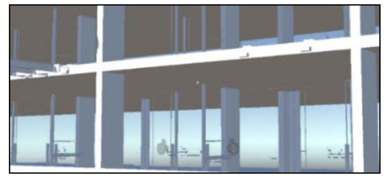


(b)

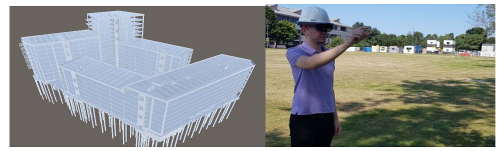
Figure 4. A comparison of the average frames per second of the different loading methods over the lapse of 120 seconds. a) Microsoft HoloLens 1 and b) Laptop Lenovo ThinkPad P52.

In Figures 4a and 4b, the scatter chart shows how the average FPS rate of the application changes over a lapse of 120 seconds using the different loading methods. As illustrated in both charts, the use of batch sizes of 5 and 10 overloads the application, dropping the FPS rate below 2 (grey and orange lines in Fig. 4a and 4b). At this point, the client application becomes unusable, therefore, we focused on the use of T1FLS with a batch size of 1 in the following comparisons.

In a high performance device, such as a laptop, the application using T1FLS runs at an average FPS rate of 17.75 (blue line in Figure 4b), while using a consistent batch size of 1, the average FPS rate is of 14.5 (yellow line in Figure 4b). In a low performance device, such as the Microsoft HoloLens 1, the application with T1FLS has an average FPS rate of 16.33 (blue line Figure 4a) and using a batch size of 1, the average FPS rate drops to 7 (yellow line Figure 4a). The T1FLS proves to be significantly better in low performance devices by running the application at an FPS rate 2.33 times better than the best static batch. Furthermore, the T1FLS proves to be adaptable between low performance and high performance devices. The average FPS rate of the T1FLS dropped 8% when switching from a high performance to a low performance device. In this same scenario the small batch method dropped 51.7%.



(a)



(b) (c)

Figure 5. a) First-person view of BIM objects in the client app. b) Third-person view of the BIM model. c) Person using Microsoft HoloLens to visualize the BIM model outdoors

Figure 5 illustrates the client application. Here, Fig. 5a shows the user's view after some 3D BIM objects were loaded using the proposed T1FLS. The complete view of the BIM model used is shown in Fig. 5b and a user interacting with the model through the AR glasses is shown in Fig. 5c. The T1FLS does not force the download of any object if the application is not capable of handling it at that point. The disadvantage of not forcing the load process is that less objects are displayed in the same amount of time. After 120 seconds of the application running, the small batch method loaded 1145 objects in the high performance device and 657 in a low performance device. The application using FLS loaded 17.2% less objects in the laptop and 51.5% less objects in the Microsoft HoloLens 1. This disadvantage is worth considering when using a high performance device to visualize the virtual environment and information. However, in low performance devices like the Microsoft HoloLens 1, the low FPS rate (7 FPS) when using a batch of 1 limited user's interaction with the BIM model and the user could not take advantage of having more objects in display. As a result of the use of fuzzy logic in this problem, we empirically developed a set of rules of human linguistic terms that characterizes the context where the application is running [30]. The knowledge can then be transferred to other AR devices by using the same set of rules.

II. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a solution for finding the optimum number of 3D data objects (i.e. BIM models) to be loaded at each frame, improving user experience by avoiding a drop in the FPS rate of the AR application when running in low performance devices (e.g. Microsoft HoloLens 1). Most of the related work in this area seek to reduce quality of the BIM model and preload it to the application. The proposed system seeks to adapt and reduce the number of loading requests at runtime based on the capabilities of the device. This is done by using a T1FLS to calculate the optimum batch size of 3D objects to be loaded at each frame of the AR application. Based on distance between the object and the user, the memory usage level, the FPS rate and the downloading speed, the T1FLS reduced the batch size to an optimum number. This allowed the AR application to run, in low performance devices like the

Microsoft HoloLens 1, at an average FPS rate of 16.33. This is 2.33 times faster than the better rate when using static batch size.

This empirically developed solution showed an improvement in average FPS of the T1FLS compared to fixed batch size method, which makes it a better solution for implementing it in AR applications when using devices with limited resources, such as the Microsoft HoloLens 1.

For our future work, we will continue working towards building a distributed architecture for visualizing and interacting with multiple 3D objects during onsite tasks through AR applications. In this paper we focused on the loading process of the 3D objects. Next steps will look on how to handle the unloading of objects that become irrelevant for the scene when the user changes spatial location while using advanced learning systems to learn and optimise the FLS parameters from data.

ACKNOWLEDGMENTS

We would like to thank Galliard Homes for the assistance provided with BIM data.

REFERENCES

[1] G. Schall et al., “Handheld Augmented Reality for underground infrastructure visualization,” Personal and Ubiquitous Computing, vol. 13, no. 4, pp. 281–291, 2009.

[2] J. Zak and H. Macadam, “Utilization of building information modeling in infrastructure’s design and construction,” in IOP Conference Series: Materials Science and Engineering, 2017, vol. 236, no. 1.

[3] A. Akcamete, B. Akinci, and J. H. Garrett, “Potential utilization of building information models for planning maintenance activities,” in Proceedings of the International Conference on Computing in Civil and Building Engineering, 2010.

[4] A. Pena-Rios, H. Hagras, G. Owusu, and M. Gardner, “Furthering Service 4.0: Harnessing Intelligent Immersive Environments and Systems,” IEEE Systems, Man, and Cybernetics Magazine, 2018.

[5] X. Wang and P. E. D. Love, “BIM + AR: Onsite information sharing and communication via advanced visualization,” in Proceedings of the 2012 IEEE 16th International Conference on Computer Supported Cooperative Work in Design, CSCWD 2012, 2012.

[6] The National Building Specification, “What is Building Information Modelling (BIM)?,” 2016. [Online]. Available: https://www.thenbs.com/knowledge/what-is-building-information-modelling-bim. [Accessed: 02-Jul-2019].

[7] Associated General Contractors of America, “The Contractor’s Guide to BIM.” [Online]. Available: https://www.engr.psu.edu/ae/thesis/portfolios/2008/tjs288/Research/AGC_GuideToBIM.pdf. [Accessed: 18-Jun-2019].

[8] Cooperative Research Centre for Construction Innovation, “Adopting BIM for facilities management,” 2007. [Online]. Available: http://www.construction-innovation.info/images/CRC_Dig_Model_Book_20070402_v2.pdf. [Accessed: 18-Jun-2019].

[9] M. Issitt, “Building Information Modeling (BIM),” Salem Press Encyclopedia. 2019.

[10] T. A. Teo and K. H. Cho, “BIM-oriented indoor network model for indoor and outdoor combined route planning,” Advanced Engineering Informatics, vol. 30, no. 3, pp. 268–282, 2016.

[11] C. Sydora and E. Stroulia, “Augmented reality on building information models,” in 2018 9th International Conference on Information, Intelligence, Systems and Applications (IISA), 2018, pp. 1–4.

[12] A. Peña-Rios, H. Hagras, M. Gardner, and G. Owusu, “A type-2 fuzzy logic based system for augmented reality visualisation of georeferenced data,” IEEE International Conference on Fuzzy Systems, 2018.

[13] A. Peña-Rios, H. Hagras, M. Gardner, and G. Owusu, “A fuzzy logic based system for geolocated augmented reality field service support,” in IEEE International Conference on Fuzzy Systems, 2017.

[14] X. Wang, P. Ed Love, and P. R. Davis, “BIM + AR: A framework of bringing BIM to construction site,” Construction Research Congress 2012: Construction Challenges in a Flat World, Proceedings of the 2012 Construction Research Congress, pp. 1175–1181, 2012.

[15] J. Ren, Y. Liu, and Z. Ruan, “Architecture in an Age of Augmented Reality: Applications and Practices for Mobile Intelligence BIM-based AR in the Entire Lifecycle,” DEStech Transactions on Computer Science and Engineering, pp. 664–674, 2017.

[16] Y. C. Lin and Y. C. Su, “Developing mobile- and BIM-based integrated visual facility maintenance management system,” The Scientific World Journal, vol. 2013, 2013.

[17] X. Liu, N. Xie, K. Tang, and J. Jia, “Lightweighting for Web3D visualization of large-scale BIM scenes in real-time,” Graphical Models, vol. 88, pp. 40–56, 2016.

[18] P. Dahne and J. N. Karigiannis, “Archeoguide: System architecture of a mobile outdoor augmented reality system,” in Proceedings - International Symposium on Mixed and Augmented Reality, ISMAR, 2002, pp. 263–264.

[19] C. Lynch, H. Hagras, and V. Callaghan, “Embedded interval type-2 neuro-fuzzy speed controller for marine diesel engines,” in Proceedings of the International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, 2006, pp. 1340–1347.

[20] H. Hagras, V. Callaghan, M. Colley, and M. Carr-West, “A Fuzzy-genetic based embedded-agent approach to learning and control in agricultural autonomous vehicles,” in Proceedings - IEEE International Conference on Robotics and Automation, 1999, no. 2, pp. 1005–1010.

[21] J. Andreu-Perez, F. Cao, H. Hagras, and G. Z. Yang, “A Self-Adaptive Online Brain-Machine Interface of a Humanoid Robot Through a General Type-2 Fuzzy Inference System,” IEEE Transactions on Fuzzy Systems, vol. 26, no. 1, pp. 101–116, 2018.

[22] H. Hagras, M. Colley, V. Callaghan, and M. Carr-west, “Online Learning and Adaptation of Autonomous Mobile Robots for Sustainable Agriculture,” Autonomous Robots, vol. 13, pp. 37–52, July 2002.

[23] A. Starkey, H. Hagras, S. Shakya, and G. Owusu, “A multi-objective genetic type-2 fuzzy logic based system for mobile field workforce area optimization,” Journal of Information Sciences, vol. 329, pp. 390–411, 2016.

[24] H. Hagras, V. Callaghan, M. Colley, G. Clarke, and H. Duman, “Online Learning and Adaptation for Intelligent Embedded Agents Operating in Domestic Environments book,” in In the book entitled “Fusion of Soft Computing and Hard Computing for Autonomous Robotic Systems” (Eds: C.Zhou, D. Maravall and D. Ruan), in the International Series “Studies in Fuzziness and Soft Computing“, vol. 116, pp. 293–323, November 2002.

[25] J. M. Mendel, Uncertain Rule-Based Fuzzy Logic Systems Introduction and New Directions, 1st editio. Prentice-Hall, Inc., 2001.

[26] H. Hagras and C. Wagner, “Introduction to Interval Type-2 Fuzzy Logic Controllers - Towards Better Uncertainty Handling in Real World Applications,” The IEEE Systems, an and Cybernetics eNewsletter, no. 27, 2009.

[27] A. Sakalli, T. Kumbasar, E. Yesil, and H. Hagras, “Analysis of the performances of type-1, self-tuning type-1 and interval type-2 fuzzy PID controllers on the Magnetic Levitation system,” in Proceedings of the 2014 IEEE International Conference on Fuzzy Systems, 2014.

[28] M. Antonelli, D. Bernardo, H. Hagras, and F. Marcelloni, “Multiobjective Evolutionary Optimization of Type-2 Fuzzy Rule-Based Systems for Financial Data Classification,” IEEE Transactions on Fuzzy Systems, vol. 25, no. 2, pp. 249–264, 2017.

[29] Microsoft, “App Quality Crtieria,” 2018. [Online]. Available: https://docs.microsoft.com/en-us/windows/mixed-reality/app-quality-criteria. [Accessed: 18-Jun-2019].

[30] A. Steventon and S. Wright, “Programming iSpaces: A Tale of Two Paradigms,” in Intelligent Spaces: The Application of Pervasive ICT, 2005, pp. 389–421.