International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Optimizing Cross Platforms Regression Testing For VR Applications

Komal Jasani

QA Engineering Lead
Union City, California USA
komal_jasani@yahoo.com

Abstract
Performing optimal regression tests across platforms remains crucial for providing uniform performance and functionality between Oculus devices with HTC Vive and PlayStation VR. The research investigates three main approaches, including automation along with compatibility frameworks and AI-based testing, which help solve performance variability challenges coupled with hardware fragmentation issues. Developers who use Unity Test Framework alongside Unreal Engine's Automation Tool shorten testing periods while increasing precision besides boosting user experience. Future directions for VR-specific testing emerge from the present analysis, which reveals existing practice shortcomings while identifying metrics to reduce latency and stop motion sickness to reach more immersive experiences.

Keywords: Cross-Application Testing, Virtual Reality, Applications Of Virtual Reality, Testing Efficiency Improvement, VR Hardware Compatibility

I. INTRODUCTION

An overview of Virtual Reality (VR) application development hurdles emphasizing cross-platform compatibility sets the introduction framework. With increasing VR technology integration across gaming and healthcare alongside education and entertainment, developers focus on application compatibility for Oculus and HTC Vive and PlayStation VR platforms. The article shows how regression testing sustains seamless functionality together with consistent user experience between all platforms. The research also presents an aim of optimizing regression testing methods to deliver efficiency together with shorter testing durations while achieving high-quality visual performance. The research tackles these VR testing difficulties in order to improve existing VR application quality features, along with designing valuable developer solutions.

II. OVERVIEW OF VR APPLICATION DEVELOPMENT

Over the past decade, Virtual Reality (VR) has experienced rapid expansion which now touches many business fields including healthcare, gaming, education and entertainment markets. Thanks to improved VR hardware such as Oculus and HTC Vive and PlayStation VR products, now most people can access immersive experiences [1]. The quick development of virtual reality hardware devices yielded many hardware configurations which now differ in tracking precision and motion detection and graphics processing capabilities. VR developers confront the essential task of delivering uniform application performance when their products run across different hardware configurations. The creation of specialized development strategies become necessary for platforms because their unique features demand independent attention to distinct APIs and hardware architectures and user input systems [2].

VR devices use distinct hardware-specific APIs that produce different behavioral paths for software applications, so developers struggle to make unified applications deliver stable performance on all platforms. Applications require optimization for their individual performance characteristics by developers who must use cross-platform development frameworks to bridge platform differences between devices. Industrial design requirements linked to consistency and quality play an essential role in securing unified high-quality VR experiences for all hardware platforms [1] [3].

III. CHALLENGES IN REGRESSION TESTING FOR VR

Before releasing changes to a VR application, developers conduct regression testing to confirm updates will not trigger new bugs while maintaining platform-wide functional performance [2]. Testing VR applications faces multiple unique obstacles because of the way these applications' function. Across various platforms, hardware-specific APIs create challenges for uniform performance outcomes because of the high dependence. Testing of VR technology requires special considerations since devices such as Oculus and HTC Vive depend on unique motion sensors alongside separate graphical rendering methods and control systems according to reference [3].

Performance problems, including frame rate changes and latency and rendering inconsistencies, frequently affect applications in virtual reality software environments. Different platforms trigger performance-related challenges that affect user experience negatively where complete testing is absent. User interactions in VR spaces extend beyond traditional controls because they include motion tracking and haptic feedback and gestural actions. Specialized testing strategies become necessary for smooth platform performance because of the unique interactions that create complex testing situations which are challenging to replicate [2] [3].

Functional correctness alongside user experiences testing requirements forms the core elements of VR evaluation. Technical tests must evaluate the user experience regarding system performance alongside their interactive comfort in VR settings. Mechanisms such as motion sickness alongside physical discomforts from inadequate system performance and latency problems actively work against creating a positive user experience. The challenge arises from maintaining detailed functional verification and superior user satisfaction in spectacular ways when performing regression tests for VR applications [4].

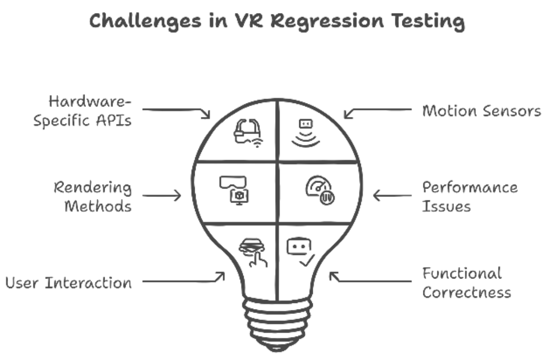


Fig 1: Challenges in VR Regression Testing

LITERATURE REVIEW

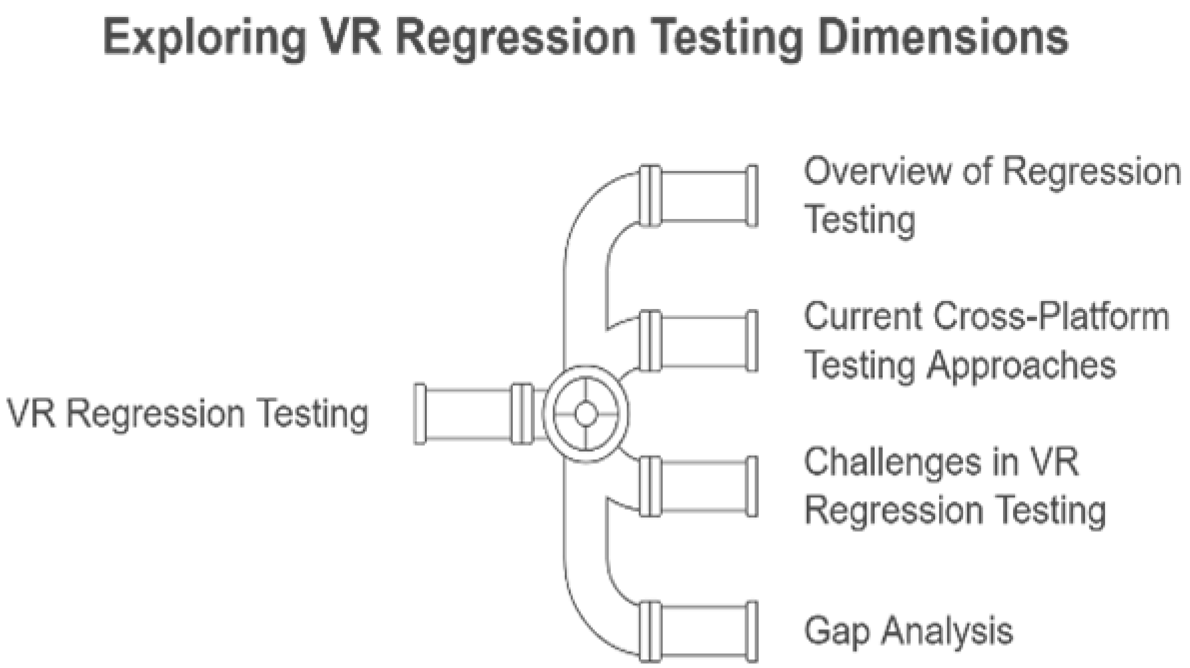
The literature review comprehensively examines how regression testing works within Virtual Reality application development by analyzing cross-platform functionality and developer obstacles and established testing approaches. The first segment provides basic information about regression testing alongside its importance for software stability after presenting dedicated requirements for VR application testing. This review evaluates present methods of conducting cross-platform testing for VR applications by discussing relevant testing tools and platforms which enable smooth VR application functionality across different hardware platforms. This study examines VR hardware peculiarities and interaction system issues alongside the current testing methodologies' limitations. Through a gap analysis, this report shows how standardized frameworks and enhanced automation technologies must address the unique requirements during VR regression testing.

A) OVERVIEW OF REGRESSION TESTING

The software development lifecycle requires regression testing, which verifies system improvements alongside every change in order to prevent newly discovered bugs and secondary effects. Traditional software development uses regression testing to make sure former workings show no disruption following different code inputs. Virtual Reality applications bring added complexity to regression testing because of their exceptional framework while also handling the hardware-software interaction dynamics [1].

Virtual Reality application needs precise implementation because of its dependency on rapid graphical processing and real-time performance measurements combined with user responsiveness demands. Implementing regression testing in VR must center on performance analysis for rendering and measuring frame rates together with execution latency and the delay of user interface components. The scope in testing traditional interfaces encompasses only functional correct output while interface design, while VR testing contains three essential dimensions of spatial interaction and visual accuracy alongside user well-being. A key requirement for preventing user discomfort through motion sickness during VR experiences is having both low latency and high frame rate performance. The quality of user experiences together with the application's effectiveness depends strongly on these key aspects [2][3].

The increasing complexity of VR applications creates a demanding situation for full-scale regression testing protocols. Specialized frameworks for VR-testing must develop to support the demands of this unique hardware as VR experiences require maintenance of interactive immersion between standalone headsets and tethered systems [4]. Standard regression testing methods regularly require adaption or expansion to properly address the implementation demands of VR performance and user interaction testing.



B) CURRENT CROSS-PLATFORM TESTING APPROACHES

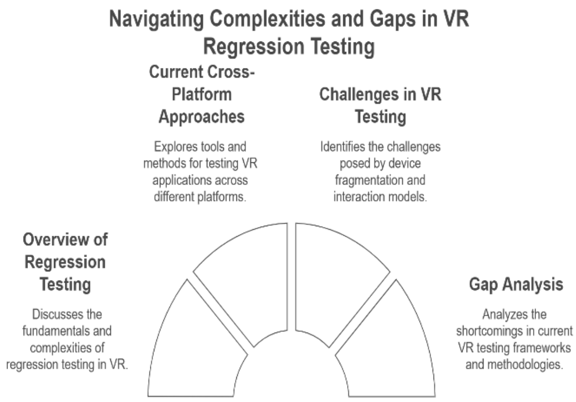
The wide variety of VR devices existing on the market makes cross-platform testing stand as a major development challenge for VR application developers. The VR platforms Oculus and HTC Vive and PlayStation VR operate with independent hardware designs, together with different graphic rendering approaches and input chair systems and entry programming interfaces. The delivery of outstanding app performance across multiple devices represents a significant development hurdle for developers. Various automated testing tools and frameworks were developed to solve this testing problem, enabling projects to finish tests more swiftly and minimize human mistakes. Unity offers one of the most applied frameworks for testing VR applications while maintaining its Test Framework. As a leading VR game, engine unity enables developers to access the Unity Test Runner alongside the Test Framework for automated testing of VR applications. Unity's testing framework includes support for two testing approaches: units and integrations, which enables developers to test game logic and VR environment interactions and performance difficulties. Through device integration between the Unity Test Framework and Oculus, or HTC Vive hardware developers can execute tests focused on frame rates and latency measurements and interaction response analysis [5]. The challenges related to maintaining cross-platform VR compatibility surface particularly frequently within Unity's Test Framework environment.

Testing software Appium VR serves mobile VR applications by providing cross-platform automation, which ensures compatibility with both Android and iOS devices. The primary purpose of Appium VR centers on mobile app testing, but it extends capabilities for monitoring VR applications on Android and iOS devices where it executes automated UI tests across hardware systems. The dual Android and iOS compatibility of Appium VR makes possible a single tool for testing applications that run on mobile VR devices, such as Oculus Quest, which operates on the Android platform. The platform serves its purpose effectively with specific VR applications, but its performance remains restricted from conducting complex motion tracking or delivering haptic feedback for immersive experiences [7]. Through abstraction layers, developers can enhance platform compatibility and lower their test costs by reducing requirements for device-specific adaptations [8].

C) CHALLENGES IN VR REGRESSION TESTING

Regression testing in virtual reality (VR) faces several significant challenges, particularly because of device fragmentation. Different VR platforms have multiple hardware configuration choices, including tracking methods controllers and input systems, along with display technologies. Each VR device operates with different tracking approaches as the Oculus Quest executes inside-out and the HTC Vive requires external sensors, while platforms select between hand controllers or full-body motion tracking methods as input options. Inconsistent performance between different devices becomes challenging because of design differences between platforms. Different hardware platforms introduce complexities during testing because they possess separate functions and show individual variances, including unique refresh rates and screen resolutions and input properties. The different performance capacities between high-end PC VR headsets such as HTC Vive and standalone models, such as Oculus Quest, mandate specific optimization measures for distinct capabilities but should cause platform uniformity [5][6].

Testing interaction models within VR represents a major hurdle during VR regression testing because of their complex nature. VR applications work with challenging user inputs, including hand tracking and gestures with controllers, yet these interactions prove difficult to test because these systems operate differently between various platforms. The expected immediate response requirements for VR interfaces result in delayed reactions, which potentially cause users discomfort alongside motion sickness development. Developers need methods to replicate authentic user motion patterns because they must verify the application reactions work similarly on diverse hardware platforms. Complexity estimating testing times is made more difficult when including supplementary sensory inputs such as haptic feedback. Specialized frameworks and methodologies developed by developers work to handle these challenges when creating VR applications which produce smooth user experiences throughout multiple devices and platforms [7][8][9].



D) GAP ANALYSIS

Standardized testing frameworks that specifically target Virtual Reality applications are missing from present VR regression testing. VR testing needs specialized tools beyond the current infrastructure because existing tools such as Unity Test Framework and Unreal's Automation Tool focus on generic testing but neglect the advanced features of VR applications, including motion tracking and stereoscopic rendering [5][6]. The lack of automated strategies to conduct VR hardware-in-loop testing with real device interaction emerges as a major gap in existing solutions. The adoption of automation through traditional development methods shows signs of inadequacy in VR testing because present tools lack complete automation capability for hardware-in-loop tests that ensure actual performance outcomes. The current automatic testing framework's restrictions force teams to depend more on manual tests while those processes demand extensive manual labor and produce frequent human mistakes [9]. Additional research on standardized testing frameworks alongside improvements to hardware-in-loop testing automation must be explored to ensure efficiency and reliability for VR applications during their technological growth.

FUTURE WORK

By using artificial intelligence (AI) to analyze previous test data and detect performance patterns researchers can forecast regression problems. Self-operated AI testing tools show promise to predict avoidance points reproducing outcomes which enhances testing protocols and squeezes out manual-based testing at the same time. The addition of VR-specific testing libraries designed for VR application requirements such as motion tracking requirements would establish enhanced testing environments which are precise and robust for these requirements. The designed libraries must achieve seamless integration with multiple VR platforms to perform tests which accurately replicate actual user experiences [1][2]. The innovations would let developers handle problems in advance and create better user experiences.

The future expansion of VR regression testing depends heavily on automation methods. The current VR testing automation depends on Unity Test Framework alongside Unreal's Automation Tool yet researchers continue to explore ways to improve automated hardware-in-loop testing interactions. Future verification processes of performance metrics should incorporate real-time simulation tests of dynamic environments and real-time user interactions to run automated tests of VR applications [6]. The industry needs a refined system that allows effortless support for different VR hardware platforms to maintain uniform testing procedures among multiple devices. OpenXR-based universal testing frameworks enable developers to establish standardized testing solutions which eliminate device-specific code when interfacing with multiple VR systems [7][8]. Research needs to create additional testing metrics that examine virtual reality-specific challenges including how to prevent sickness during motion and materials related to frame rate stability and latency measurements in order to develop user-friendly VR applications. The improvements created new standards that enhanced both scalability and reliability of VR regression testing in the quickly advancing virtual reality domain.

ACKNOWLEDGEMENTS

All supporters of this project including the developers and researchers and contributors deserve my heartfelt thanks. The research significantly relied on tools from Unity and Unreal Engine development teams for which I express my genuine appreciation. This work was funded by the agencies supporting this research and directed by the valuable feedback provided by colleagues. Multiple individuals has brought significant advancement to this study dedicated to Virtual Reality regression testing.

DISCUSSION

When developing VR, the utilization of efficient cross-platform testing methods proves essential in shortening testing periods while aspiring to maintain equivalent device performances. VR optimization techniques require the implementation of platform-agnostic elements through OpenXR abstraction layers, which eliminates the requirement for writing a distinct code for each device. The testing burden reduces and functional consistency between platforms enhances by these methods [5]. Both Unity game engines and Unreal game engine feature testing efficiency tools which help developers automate solutions. Through its automated feature set, the Test Framework from Unity and the Automation Tool from Unreal developers can execute simultaneous platform tests from customized testing scenarios which enhance testing operations. Efficient implementation of these tools demands strategic alignment with other testing solutions to attain total system coverage [3].

The optimization of regression testing practices in VR applications produces significant benefits for product delivery speed and user satisfaction. The shortening of testing intervals through quick detection and fix rates enhances both update frequency and reduces new feature wait times. Ramifications of the quickly evolving VR market call for this approach, which ensures proper device and platform support. The optimization process removes bugs from different platforms that cause rendering glitches and latency problems, resulting in better user experience. Unlike problems that may occur from unpredictable device performance, developers can prevent motion sickness while protecting frame rates, as this enhances VR immersion [4].

Manual testing continues to provide value by evaluating subjective user comfort levels and physical hardware interaction functionality that automatic tools do not assess efficiently. Through manual testing techniques, human evaluators perform actual-use movement simulations and check ergonomic components while locating automated detection gaps [6]. Testing through human labor is both terribly slow and error-prone, particularly for testing on multiple platform types. The automated resources inside Unity and Unreal Engine have developed an enhanced ability to execute recurring verifications of frame rates and basic interaction times and system latency, which empowers testers to address complex operational challenges first. Automation boosts work efficiency, yet existing technology lacks complete capabilities to mirror user comfort and real-time motion tracking accurately [7].

The progress made in VR regression testing faces ongoing obstacles while coping with resource limitations to reach a level of 100% coverage. Testing becomes difficult because users face obstacles from the wide range of hardware performance abilities and different control systems between devices across platforms. Although abstraction layers provide help, developers perform edge case testing through both automated systems and human testers while facing increased overall development duration [6]. Actual hardware testing must continue because simulations may cannot reveal certain problems, yet the high costs of maintaining different VR devices make broad testing unattainable. Ongoing hardware evolution presents developers with logistical hurdles which create bottlenecks that limit extensive testing [5]. Continuous advancements in testing tools alongside better strategies must defeat existing limitations, which will enable high-quality experiences across multiple platforms.

CONCLUSION

The development quality of VR applications suffers significant improvement through optimized regression testing, which allows users to experience high-quality features across multiple platforms. Testing becomes more efficient possible by applying standardized frameworks OpenXR and automation tools found inside game engines Unity and Unreal Engine thus developers can cut testing time without compromising thorough platform coverage. The optimized framework identifies and fixes bugs more quickly, which speeds up the development rate while providing enhanced user experiences. Frame rate inconsistencies and latency problems show the necessity for improved regression testing approaches [5].

Optimized regression testing frameworks produce consequences which reach further than enhanced efficiency metrics. The optimized system promotes quicker software creation together with uniform app functionality across devices, which together push advanced innovation throughout the VR sector. The rise in new VR devices alongside expanding cross-platform application needs will make testing tools indispensable for assuring the reliability and quality of VR content. Implementing effective testing allows VR developers to advance their immersive content with freedom from platform or hardware restrictions. Ongoing testing strategy improvements advance both present applications and materialize opportunities for enhanced user interactions and individualized virtual environments in VR development [4][5].

The successful development of cross-platform testing capabilities depends on extreme urgency between developers, researchers, and tool creators. The continued evolution of virtual reality technology demands priority attention to the development of universal testing frameworks and testing tools. To overcome hardware fragmentation restrictions and limited availability of testing resources and emerging VR technologies, researchers must collaborate with developers. Many improved testing solutions can be built for the VR industry through open platforms, which enable research sharing and ongoing development of open-source tools and hardware testing collaboration. The collaborative works these efforts establish enables development streamlining and creates future engineering possibilities for innovative multi-platform accessible virtual reality solutions [6][7].

REFERENCES

[1] Atluri, H., & Thummisetti, B. S. P. (2022). A Holistic Examination of Patient Outcomes, Healthcare Accessibility, and Technological Integration in Remote Healthcare Delivery. *Transactions on Latest Trends in Health Sector, 14*(14).

[2] Bueckle, A., Buehling, K., Shih, P. C., & Börner, K. (2022). Optimizing performance and satisfaction in matching and movement tasks in virtual reality with interventions using the data visualization literacy framework. *Frontiers in Virtual Reality, 2*, 727344.

[3] Chen, X., Chang-Richards, A. Y., Yiu, T. W., Ling, F. Y. Y., Pelosi, A., & Yang, N. (2024). A multivariate regression analysis of barriers to digital technologies adoption in the construction industry. *Engineering, Construction and Architectural Management, 31(11), 4281-4307.

[4] Grande, R., Albusac, J., Vallejo, D., Glez-Morcillo, C., & Castro-Schez, J. J. (2024). Performance Evaluation and Optimization of 3D Models from Low-Cost 3D Scanning Technologies for Virtual Reality and Metaverse E-Commerce. Applied Sciences, 14 (14), 6037.

[5] Hu, M., Luo, X., Chen, J., Lee, Y. C., Zhou, Y., & Wu, D. (2021). Virtual reality: A survey of enabling technologies and its applications in IoT. Journal of Network and Computer Applications, 178, 102970.

[6] Li, S., Gao, C., Zhang, J., Zhang, Y., Liu, Y., Gu, J.,& Lyu, M. R. (2024). Less cybersickness, please: Demystifying and detecting stereoscopic visual inconsistencies in virtual reality apps. *Proceedings of the ACM on Software Engineering, 1*(FSE), 2167-2189.

[7] Li, Z., Pestourie, R., Park, J. S., Huang, Y. W., Johnson, S. G., & Capasso, F. (2022). Inverse design enables large-scale high-performance meta-optics reshaping virtual reality. *Nature Communications, 13 (1), 1-11.

[8] Ma, L., Xie, B., Liu, F., & Ma, L. (2024). A Method of Applying Virtual Reality Converged Remote Platform Based on Crawfish Optimization Algorithm to Improve ESN Network. *EAI Endorsed Transactions on Scalable Information Systems, 11 (3).

[9] Matsubara, T., Ishihara, S., Nagamachi, M., & Matsubara, Y. (2011). Kansei Analysis of the Japanese Residential Garden and Development of a Low-Cost Virtual Reality Kansei Engineering System for Gardens. Advances in Human-Computer Interaction, 2011*(1), 295074.

[10] Nusrat, F., Hassan, F., Zhong, H., & Wang, X. (2021, May). How developers optimize virtual reality applications: A study of optimization commits in open source unity projects. In *2021 IEEE/ACM 43rd International Conference on Software Engineering (ICSE)* (pp. 473-485). IEEE.

[11] Raja, M., & Lakshmi Priya, G. G. (2021). An Analysis of Virtual Reality Usage through a Descriptive Research Analysis on School Students' Experiences: A Study from India. *International Journal of Early Childhood Special Education, 13 (2).

[12] Rukhiran, M., Boonsong, S., & Netinant, P. (2024). Sustainable Optimizing Performance and Energy Efficiency in Proof of Work Blockchain: A Multilinear Regression Approach. *Sustainability, 16*(4), 1519.

[13] Yu, D., Liang, H. N., Lu, X., Fan, K., & Ens, B. (2019). Modeling endpoint distribution of pointing selection tasks in virtual reality environments. ACM Transactions on Graphics (TOG), 38(6), 1-13.

[14] Zhang, H., Lee, S., Lu, Y., Yu, X., & Lu, H. (2022). A survey on big data technologies and their applications to the metaverse: Past, current and future. Mathematics, 11 (1), 96.

[15] Zhang, Y., Liu, H., Kang, S. C., & Al-Hussein, M. (2020). Virtual reality applications for the built environment: Research trends and opportunities. *Automation in Construction, 118*, 103311.

[16] Zheng, J., Du, L., Deng, X., Zhang, L., Wang, J., & Chen, G. (2022). Efficacy of virtual reality techniques in cardiopulmonary resuscitation training: protocol for a meta-analysis of randomised controlled trials and trial sequential analysis. *BMJ Open, 12 (2), e058827.

[17] Bernaschina, C., Fedorov, R., Frajberg, D., & Fraternali, P. (2017). A framework for regression testing of outdoor mobile applications. *2017 IEEE/ACM 4th International Conference on Mobile Software Engineering and Systems (MOBILESoft)*, 201-205. DOI: 10.1109/MOBILESoft.2017.16

[18] Rao, S. N., Prapulla, S. B., Shobha, G., Hariprasad, S., Gupta, M., & Reddy, S. A. (2019). Using virtual reality to boost the effectiveness of brain-computer interface applications. *2019 IEEE 16th India Council International Conference (INDICON)*, 1-4. DOI: 10.1109/INDICON47234.2019.9029060

[19] Carneiro, J., Rossetti, R. J. F., Silva, D. C., & Oliveira, E. C. (2021). BIM, GIS, IoT, and AR/VR integration for smart maintenance and management of road networks: a review. *Journal of Infrastructure Systems*, 27(3), 04021025. DOI: 10.1061/(ASCE)IS.1943-555X.0000620

[20] Kang, H. J., Shin, J., & Ponto, K. (2020). How 3D virtual reality stores can shape consumer purchase decisions: The roles of informativeness and playfulness. *Journal of Interactive Marketing*, 49, 70-85. DOI: 10.1177/1094996820919750

[21] Chen, Y., Hu, S., Mao, H., Deng, W., & Gao, X. (2020). Application of the best evacuation model of deep learning in the design of public structures. *Image and Vision Computing*, 102, 103974. DOI: 10.1016/j.imavis.2020.103974

[22] Rane, N., Choudhary, S., & Rane, J. (2023). Leading-edge technologies for architectural design: a comprehensive review. *SSRN Electronic Journal*. DOI:10.2139/ssrn.4637891

[23] Tank, K. H., Ghanem, M. C., Vassilev, V., & Watanobe, Y. (2025). Synchronization, optimization, and adaptation of machine learning techniques for computer vision in Cyber-Physical Systems: a comprehensive analysis. DOI: 10.20944/preprints202501.0001.v1

[24] Goh, H. A., Ho, C. K., & Abas, F. S. (2023). Front-end deep learning web apps development and deployment: a review. ***Applied Intelligence***, 53(4), 4567-4583. DOI: 10.1007/s10489-023-04156-7

[25] Fu, J., Chen, Z., Chen, X., & Li, W. (2020). Sequential reinforced 360-degree video adaptive streaming with cross-user attentive network. IEEE Transactions on *Multimedia*, 22(12), 3023-3036. DOI:10.1109/TMM.2020.2977042

[26] Pinter, C., Lasso, A., Choueib, S., Asselin, M., & Fichtinger, G. (2020). SlicerVR for medical intervention training and planning in immersive virtual reality. *IEEE Transactions on Medical Robotics and Bionics*, **2**(2), 148-156. DOI: 10.1109/TMRB.2020.2991645

[27] Jaganov, T., Nnadi, C. L., & Watanobe, Y. (2024). The Learning Labyrinth: Integrating Learning Theories in VR. ***Proceedings of the 2024 5th Asia Pacific Information Technology Conference***, 45-50. DOI: 10.1145/3383210.3383215