IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, July 2026
Impact Factor: 8.2

Review of Automated Testing Techniques for Augmented Reality and Virtual Reality Applications

Shivputra Guruling Swami[1] and Dr. Mohammad Suaib[2]
[1]Research Scholar, Department of Computer Science
[2]Associate Professor, Department of Computer Science
Sabarmati University, Ahmedabad

Abstract: *Augmented Reality and Virtual Reality applications are rapidly being adopted in education, healthcare, gaming, manufacturing, military training, tourism, engineering, and other domains. Unlike conventional software applications, AR and VR systems involve three-dimensional environments, immersive interaction, spatial tracking, motion sensing, real-time rendering, multimodal input, and six-degrees-of-freedom interaction. These characteristics make conventional software testing approaches insufficient for identifying many functional, visual, interaction, usability, and performance-related defects. Automated testing has therefore emerged as an important research area for improving the reliability, scalability, and efficiency of AR and VR application development.*

Keywords: Augmented Reality, Software Testing, Model-Based Testing, Machine Learning, Test Generation, Scene Testing, Visual Testing

I. INTRODUCTION

Augmented Reality and Virtual Reality represent important forms of Extended Reality (XR) technology that provide users with interactive and immersive digital experiences. AR combines virtual content with the physical environment, whereas VR creates a predominantly computer-generated environment in which users interact through head-mounted displays, controllers, sensors, cameras, and other input devices. The growing complexity of these applications has created significant challenges for software quality assurance. Traditional applications generally rely on two-dimensional graphical user interfaces and predictable input mechanisms, while AR and VR systems involve spatial relationships, object tracking, gestures, head movements, hand interactions, physical surroundings, and continuously changing visual scenes.

Consequently, traditional testing approaches cannot always adequately determine whether an immersive application behaves correctly under different spatial and interaction conditions. Research on XR testing has identified six-degrees-of-freedom interaction and other immersive characteristics as important factors that distinguish XR testing from conventional 2D graphical-interface testing. A 2025 systematic mapping study examining 34 primary studies found that automated testing was one of the most prominent research topics in XR testing, demonstrating the growing importance of automation in this area.

Automated testing is particularly valuable because manually testing every possible interaction, viewpoint, movement, environmental condition, and virtual object relationship is expensive and time-consuming. An AR or VR application may contain hundreds of possible interaction paths, and defects may only appear under specific combinations of user movement, object position, camera orientation, lighting, device configuration, or environmental conditions.

Automated techniques can repeatedly execute predefined tests, generate test inputs, explore virtual scenes, identify abnormal behaviour, compare expected and observed outputs, and evaluate visual or spatial properties. Early research on automated testing of VR application interfaces demonstrated the need to adapt standard software-engineering testing

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, July 2026
Impact Factor: 8.2

practices to immersive interaction environments. Later work developed automated functional-testing approaches specifically for VR applications, reflecting the increasing movement from manual testing toward systematic and automated fault detection.

NEED FOR AUTOMATED TESTING IN AR AND VR APPLICATIONS

The need for automated testing arises from the distinctive architecture and behaviour of AR/VR systems. First, these applications operate in three-dimensional spaces in which the relative position and orientation of objects can directly influence application behaviour. A virtual object may need to remain attached to a physical surface, respond to a user's movement, avoid another object, or appear correctly from different viewpoints. Second, AR systems depend heavily on environmental sensing and tracking. Changes in lighting, camera position, surface characteristics, device movement, and environmental geometry can affect application performance.

Third, VR applications require accurate interaction between virtual scenes and input devices, making controller movement, hand tracking, head orientation, collision, and object manipulation important testing dimensions. Fourth, visual defects may not cause application crashes but can significantly reduce user experience and functional correctness. For example, an incorrectly positioned virtual object may still allow the application to execute normally but represent a serious application defect. Automated testing can address these challenges by combining functional testing with visual analysis, spatial reasoning, machine learning, and automated interaction.

MAJOR AUTOMATED TESTING TECHNIQUES

1. Model-Based Testing

Model-based testing represents the behaviour of an AR/VR application through an abstract model consisting of states, events, actions, transitions, and expected outcomes. Automated test cases can then be generated by exploring the model. This approach is particularly useful for applications containing complex interaction sequences because the model can represent different states of a virtual scene and the transitions caused by user actions. Model-based testing can systematically cover interaction paths that may be difficult to test manually. The recent XR testing literature identifies model-based testing as one of the important techniques used in this domain, although its application remains smaller than that of machine-learning-based approaches.

2. Machine-Learning-Based Testing

Machine learning has become an important technique for automated AR/VR testing because immersive applications generate large quantities of visual and interaction data. Machine-learning models can be trained to recognize abnormal object positions, incorrect scene configurations, visual inconsistencies, and other defects. In particular, image-based approaches can analyse screenshots generated during automated execution and compare them with expected patterns. Recent mapping research identified machine-learning-based testing as the most frequently used testing technique among the reviewed XR studies, appearing in nine studies.

3. Automated Test-Input Generation

Test-input generation focuses on automatically producing different actions, positions, movements, viewpoints, object configurations, and interaction sequences. This is essential because manually specifying all possible combinations is impractical. Automated test-input generation may use random testing, combinatorial strategies, search algorithms, or intelligent agents. For VR, generated inputs may include controller movements, object selections, navigation actions, and interaction sequences. For AR, inputs may include camera movement, surface detection, object placement, and changes in physical context.

4. Automated Scene Exploration

Scene exploration techniques automatically navigate through virtual environments and interact with available objects. These methods are especially useful for VR applications where the virtual environment may contain many rooms, objects, interaction points, and possible paths. Automated agents can explore different routes and trigger interactions to discover

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, July 2026
Impact Factor: 8.2

crashes, incorrect transitions, inaccessible objects, or unexpected application states. Research has proposed automated VR scene-testing approaches that explore environments, interact with objects, and optimize interaction routes.

5. Visual and Screenshot-Based Testing

Visual testing is particularly important for AR and VR because many defects are graphical or spatial rather than purely functional. Automated screenshot analysis can identify incorrect object placement, missing objects, visual inconsistencies, rendering problems, and other scene-level defects. Neural-network-based techniques have been investigated for detecting incorrect object placement in AR applications. Such methods address an important limitation of traditional testing: the application may continue running correctly while presenting an incorrect visual result to the user.

6. Automated Functional Testing

Functional testing determines whether an application performs its intended operations correctly. In VR, automated functional testing can verify navigation, object manipulation, interaction sequences, menu operations, scene transitions, and application responses. Automated functional testing is especially useful because VR applications often contain complex interactions that are difficult to validate manually. A dedicated automated functional-testing approach for VR applications demonstrated the importance of generating and executing tests systematically rather than relying primarily on manual usability testing.

7. Search-Based Testing

Search-based testing uses optimization algorithms to identify test cases that maximize coverage or increase the probability of discovering faults. Genetic algorithms, evolutionary algorithms, and other search techniques can generate interaction sequences or spatial configurations. This approach is useful when the possible test space is very large. For example, an algorithm may search for combinations of object positions and user movements that expose collision or interaction defects.

8. Reinforcement-Learning-Based Testing

Reinforcement learning can be used to create intelligent testing agents that learn how to interact with an AR/VR environment. Instead of following only predefined scripts, an agent can learn which actions are likely to reveal new states or faults. Such techniques are promising for large and dynamic virtual environments because the agent can adapt its exploration strategy based on feedback from the application. However, reinforcement-learning-based testing requires appropriate reward functions, training environments, and reliable mechanisms for recognizing faults.

9. Automated Collision and Occlusion Testing

Collision and occlusion are important XR-specific properties. Collision testing verifies whether objects interact physically or logically as expected, whereas occlusion testing verifies whether objects correctly appear in front of or behind other objects. Automated test-data generation has been explored for identifying incorrect collision and occlusion behaviour in VR applications. These techniques are particularly important because conventional GUI testing cannot adequately represent three-dimensional spatial relationships.

10. Automated Performance and Usability Testing

Performance testing evaluates frame rate, latency, resource utilization, rendering performance, tracking responsiveness, and system stability. Automated performance testing can execute applications under controlled workloads and record performance indicators. Usability testing can also be partially automated through interaction logs, movement patterns, task completion times, and behavioural analysis. Nevertheless, human evaluation remains important for subjective dimensions such as comfort, immersion, presence, cybersickness, and overall user satisfaction.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, July 2026
Impact Factor: 8.2

COMPARATIVE REVIEW OF AUTOMATED TESTING TECHNIQUES

Testing Technique	Main Purpose	Typical Test Target	Major Advantage	Major Limitation
Model-Based Testing	Generate tests from behavioural models	States, transitions, interactions	Systematic test generation	Requires accurate models
Machine-Learning-Based Testing	Detect complex visual or behavioural faults	Images, scenes, object placement	Handles complex visual information	Requires training data
Automated Test-Input Generation	Produce diverse test cases	User actions, positions, movements	Reduces manual effort	Large input space
Scene Exploration	Explore virtual environments	VR scenes and objects	Finds unexplored interaction paths	Exploration may be expensive
Visual/Screenshot Testing	Detect visual defects	Screenshots, rendered scenes	Detects non-crashing visual faults	Sensitive to viewpoint and rendering variations
Functional Testing	Verify application functions	Interactions and workflows	Detects functional defects	Complex immersive interactions
Search-Based Testing	Optimize test-case selection	Paths, interactions, spatial states	Efficient search of large spaces	Depends on fitness function
Reinforcement Learning	Learn testing strategies	Interactive environments	Adaptive exploration	Training complexity
Collision Testing	Verify spatial interactions	Object-object relationships	Detects XR-specific defects	Requires spatial oracles
Occlusion Testing	Verify visual depth relationships	Virtual/real objects	Useful for AR realism	Environmental variation
Performance Testing	Measure system efficiency	Frame rate, latency, resource use	Supports continuous monitoring	Hardware-dependent results
Security Testing	Detect security/privacy defects	Sensors, data, interactions	Identifies XR-specific threats	Security oracles can be difficult
Unit Testing	Test individual components	Scripts, functions, modules	Fast and repeatable	Limited scene-level coverage
Integration Testing	Verify component interaction	Software and hardware components	Finds interface defects	Configuration complexity

The comparative evidence indicates that no single automated technique is sufficient for complete AR/VR quality assurance. The 2025 systematic mapping study identified 14 different testing techniques across the reviewed literature, with machine-learning-based testing and model-based testing being particularly prominent. System testing was also much more common than unit, integration, or non-functional testing, indicating that the research community has placed considerable emphasis on application-level behaviour while lower-level and broader quality attributes remain less explored.

AUTOMATED TESTING TOOLS AND FRAMEWORKS

Several research tools and frameworks have been proposed to support automated XR testing. The recent literature identifies tools such as iv4XR, VRGuide, VRTest, TESTAR, PredART, VOPA, and other research prototypes. iv4XR is

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, July 2026
Impact Factor: 8.2

notable because it provides a collection of approaches for automated XR testing, including agent-based, model-based, and reinforcement-learning-based testing. VRGuide and VRTest focus on automated exploration of VR scenes, while other tools concentrate on usability, security, visual inconsistencies, or object placement.

The development of such frameworks demonstrates an important transition from isolated testing scripts toward reusable testing infrastructures. However, tool maturity remains uneven. Some research tools are publicly available, whereas others have limited accessibility or are prototypes intended primarily for experimental evaluation. Therefore, future research should emphasize open datasets, standardized benchmarks, reusable test environments, and integration with popular development platforms such as Unity and Unreal Engine.

CHALLENGES IN AUTOMATED AR/VR TESTING

Despite significant progress, automated testing for AR/VR applications continues to face several challenges. The first major challenge is the **test-oracle problem**. A test oracle determines whether the observed application behaviour is correct. Detecting crashes is relatively straightforward, but determining whether a virtual object is positioned correctly, whether an interaction feels natural, or whether an AR object is visually aligned with the physical environment can be much more difficult. Recent research emphasizes that test generation, which includes both test inputs and test oracles, remains less explored than simple automated test execution.

The second challenge is environmental variability. AR applications operate in real-world environments where lighting, surfaces, camera position, background objects, and physical movement can change continuously. Third, hardware diversity creates additional testing complexity because AR/VR applications may behave differently across headsets, smartphones, controllers, sensors, graphics processors, and operating systems. Fourth, visual testing is affected by viewpoint, rendering conditions, device resolution, and graphical settings. Fifth, many usability and comfort characteristics remain difficult to automate because they depend on human perception.

Another challenge is the lack of standardized datasets and benchmarks. Machine-learning-based testing requires representative data containing both correct and defective examples. Without standardized datasets, comparing different testing approaches becomes difficult. The recent systematic mapping study therefore emphasizes the value of repositories containing tools and datasets for advancing XR testing research.

II. CONCLUSION

Automated testing is becoming an essential component of quality assurance for Augmented Reality and Virtual Reality applications. The immersive and spatial nature of AR/VR systems creates testing requirements that differ substantially from conventional software. Model-based testing, machine-learning-based testing, automated scene exploration, visual testing, search-based testing, reinforcement learning, automated functional testing, and collision and occlusion testing provide complementary mechanisms for identifying defects. Current research demonstrates considerable progress, but the field remains relatively young. A recent systematic mapping study identified 34 primary XR-testing studies and showed that automated testing, usability testing, security testing, and XR-specific testing are major research areas, while machine-learning-based and model-based testing are among the most frequently employed techniques.

The literature suggests that the future of AR/VR testing will depend on combining multiple automated techniques rather than relying on a single method. Machine learning can support visual fault detection, model-based approaches can provide systematic interaction coverage, intelligent agents can explore complex scenes, and automated oracles can improve fault identification. At the same time, human evaluation will continue to be necessary for subjective attributes such as comfort, immersion, presence, and usability. A mature AR/VR testing framework should therefore integrate automated functional, visual, spatial, performance, security, and usability testing within a continuous software-development lifecycle. Such integration can reduce manual testing effort, improve defect detection, increase application reliability, and support the development of safer and more dependable immersive applications.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 6, Issue 5, July 2026
Impact Factor: 8.2

REFERENCES

1. Bierbaum, A., Hartling, P., & Cruz-Neira, C. (2003). Automated testing of virtual reality application interfaces. *Proceedings of the International Immersive Projection Technology Workshop*.
2. Gu, R., Rojas, J. M., & Shin, D. (2025). Software testing for extended reality applications: A systematic mapping study. *Automated Software Engineering, 32*(2), 56.
3. Gu, U., Rojas,. (2020). Software testing for extended reality applications: A systematic mapping study. *Automated Software Engineering, 31*(1), 54.
4. Souza, A. C. C., Nunes, F. L. S., & Delamaro, M. E. (2018). An automated functional testing approach for virtual reality applications. *Software Testing, Verification and Reliability, 28*, e1690.
5. Souza, P. R. (2020). An automated functional testing approach for virtual reality applications. *Software Testing, Verification and Reliability, 26*, e1680.