WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Web-Based AR System for Real-Time 3D Model Projection

[1]Mr. Sri Vallinayagam V,[2] Dr. R. Tino Merlin,[3] Mr. Sherbin C

[1 2 3] Computer Science and Engineering, Francis Xavier Engineering College, Tirunelveli – Tamil Nadu – India

ABSTRACT:

The design, implementation, and preliminary assessment of a new web-based Augmented Reality (AR) system for the real-time projection of interactive 3D models onto physical settings are described in this study. The system seeks to give a smooth and cross-platform augmented reality experience by using the accessibility and usability provided by AR.js, a lightweight JavaScript package that enables both marker-based and markerless AR within common web browsers. We explore the details of loading and managing 3D models, the real-time tracking features that AR.js uses, and the rendering pipeline that precisely superimposes virtual items on the user's perspective. We also look at possible applications in product visualization, education, and captivating interactive displays. Our prototype's initial results reveal that this strategy is feasible, exhibiting real-time 3D model projection on a variety of web-enabled devices with respectable levels of responsiveness and accuracy. We also go over preliminary findings about user interaction and performance.

Keywords: Augmented Reality (AR), WebAR, AR.js, Real-Time 3D Projection, Interactive 3D, Marker Tracking, Markerless Tracking, WebGL Rendering, JavaScript Framework, Cross-Platform AR.

Introduction:

Augmented reality, or AR, has become a potent paradigm for fusing digital data with the real world, providing more engaging and interactive experiences. Historically, the implementation of AR apps required platform-specific software development and specialized hardware, which hindered their mainstream acceptance. Web-based augmented reality (AR) has been made possible by the development of web technologies, which allow consumers to access AR experiences directly through their current web browsers on a variety of devices, greatly expanding accessibility.[1]

This study discusses the challenge of presenting and interacting with complex 3D models in real-time within an online augmented reality environment. The seamless integration of virtual 3D content into the user's real-world surroundings opens up a vast array of potential uses. Imagine students researching the intricate structure of a molecule superimposed on their workstation, or potential buyers visualizing a piece of furniture in their living area before making a purchase[2].

Our work focuses on using the open-source JavaScript framework AR.js, which is well-known for its effectiveness and ease of use for developing web-based augmented reality apps. Both marker-based tracking—which makes use of pre-established visual markers—and markerless tracking—which depends on examining environmental features—are supported by AR.js. The architectural design, implementation details, and initial evaluation of a system that uses AR.js to accomplish the smooth projection and real-time manipulation of 3D models in a web browser context are described in this paper. We also briefly discuss the great prospects and inherent challenges of creating a system of this kind for widespread accessibility[3].

Literature Survey:

The Evolution of Web-Based Augmented Reality:

As pioneers like Sutherland envisioned with his ultimate display design, augmented reality has evolved from early, frequently unwieldy, hardware-dependent systems to the more approachable world of web-based platforms. With a focus on accurate 3D registration, real-time interaction, and the smooth merging of virtual and real elements, foundational research developed the fundamentals of augmented reality. This change has been largely attributed to the emergence of potent web technologies. In particular, the advent of WebGL has made it possible to generate 3D visuals at a high speed within web browsers, creating the potential for aesthetically stunning augmented reality experiences. A major step in offering a consistent interface for accessing AR and VR functions across different web browsers and devices, as well as expediting the development process for WebAR applications, is the creation and standardization of the WebXR Device API. A key component of many WebAR projects, complementary libraries such as three.js have given developers powerful capabilities for generating and modifying 3D scenes in the online environment.[8]

AR.js: A Key Enabler for Accessible WebAR:

An important and extensively used open-source JavaScript library designed especially for generating augmented reality experiences within web browsers is AR.js. Its support for various tracking approaches, lightweight design, and ease of integration into pre-existing web projects are its main advantages. Notably, AR.js supports a variety of markerless tracking methods in addition to marker-based tracking, which depends on the accurate and stable anchoring of virtual content through the real-time detection and tracking of predefined visual markers. By allowing augmentation based on the identification of certain target images or the analysis of environmental natural elements, these markerless techniques provide more flexibility. Numerous projects and applications in a variety of fields have adopted AR.js due to its adaptability and accessibility. These include interactive educational tools that make abstract concepts come to life, useful visualization solutions that let users see products in their natural environments, and captivating marketing campaigns that provide innovative interactive experiences straight through web browsers.[4]

Understanding Marker-Based and Markerless Tracking in WebAR :

One of AR.js's core features, marker-based tracking, offers a dependable way to register virtual objects in the actual world with accuracy and stability. This method depends on the identification and ongoing monitoring of particular, recognizable visual indicators. Accurate overlay of 3D models and other virtual information is made possible by the system's ability to precisely calculate a marker's position and orientation once it has been recognized in the camera stream. The user experience may be limited by the fact that marker-based tracking necessitates the presence and unhindered visibility of physical markers, even while it offers high levels of accuracy and robustness in controlled conditions where markers are consistently visible. However, markerless tracking methods, which are also enabled by AR.js, seek to get around this restriction by identifying certain target images or by examining natural elements in the camera's field of view. Since these techniques don't depend on the positioning of actual markers, they provide a more fluid and natural user experience. Markerless tracking, however, frequently poses more technical difficulties in terms of computing expense, reliance on the environment (such as lighting and feature availability), and attaining the same degree of accuracy and stability as properly designed marker-based tracking. The goal of ongoing research and development in marker-based and markerless tracking algorithms is to increase their resilience, efficiency, and performance for a greater variety of augmented reality applications.[5]

Positioning the Current Research within the WebAR Landscape:

This study aims to expand on the well-established framework of web-based augmented reality, concentrating on the possible interactive manipulation and real-time projection of 3D models in an AR environment that is powered by the AR.js package. In order to provide important insights for the continued development and comprehension of accessible WebAR experiences, this work will carefully examine the capabilities and intrinsic limitations of AR.js in this context, with a focus on the smooth integration and real-time rendering of 3D content as well as the application of user-friendly interaction techniques. In addition to exploring the possibility of improving user engagement through interactive features, this study aims to fill any existing knowledge gaps about the best practices for achieving responsive and fluid real-time 3D model projection within the limitations of web browser environments. In the conclusion, our work seeks to provide guidance for the creation of more reliable and intuitive web-based augmented reality applications in the future for a greater variety of real-world applications.[6]

Flow chart:

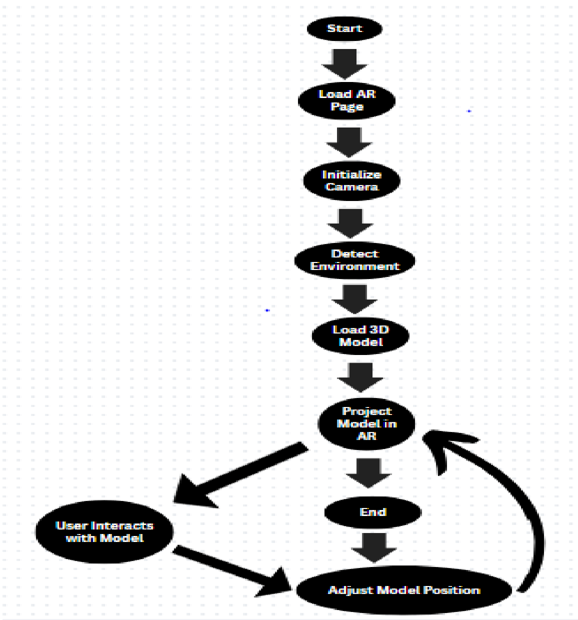

Figure 1

Proposed Methodology:

This research proposes a web-based Augmented Reality system for the real-time projection of interactive 3D models onto physical environments, leveraging the AR.js library. The methodology encompasses the following stages:

System Architecture Design:

- A client-side architecture will be adopted, where all AR processing and rendering occur within the user's web browser. The system will consist of modules for:
- **3D Model Loading and Management:**
  Handling the loading and preparation of 3D models in common web-compatible formats (e.g., glTF).
  Camera Access and Video Stream Handling:
  Utilizing the browser's WebRTC API to access the device camera feed.
- **Tracking Implementation (Marker-Based and Optional Markerless):**
  Integrating AR.js for real-time tracking using both marker-based techniques (for controlled scenarios) and optionally exploring markerless techniques (e.g., image tracking) for more flexible interactions.[10]
- **3D Scene Management and Rendering:**
  Employing three.js, integrated within AR.js, to manage the virtual 3D scene and render the loaded models onto the camera feed using WebGL.
- **User Interface and Interaction:**
  Developing a simple web interface for model selection and potentially implementing basic interaction controls (e.g., scaling, rotation) using JavaScript event listeners.[7]
- **Implementation and Development:**
  The system will be developed using HTML, CSS, and JavaScript. AR.js will be integrated for handling camera access, tracking, and basic rendering setup. Three.js will be utilized for advanced 3D model manipulation and rendering within the AR scene. For marker-based tracking, custom AR markers will be generated and used for evaluation. If markerless tracking is explored, specific target images will be selected for testing. Basic user interaction functionalities (if included) will be implemented using JavaScript event handling for touch and mouse inputs.

***System Evaluation*:**

- **Tracking Accuracy and Stability**
  Quantitative measurements of positional and rotational error of the projected 3D model relative to the tracked marker (or target image). Qualitative assessment of tracking stability under varying conditions (lighting, movement).[9]
- **Rendering Performance:**
  Measurement of frames per second (FPS) achieved on different devices (desktop, mobile) and with varying 3D model complexities. Analysis of CPU and GPU usage.
- **System Responsiveness (with interaction):**
  Measurement of the latency between user input and the corresponding change in the projected 3D model. Qualitative assessment of the smoothness of interactions.
- **User Experience:**
  Gathering user feedback on ease of use, intuitiveness of interaction, and overall satisfaction.
- **Testing Environment:**
  The system will be tested on a range of devices, including desktop computers with webcams and mobile devices (smartphones, tablets) with integrated cameras, using modern web browsers that support WebGL and WebRTC.

Results & Discussion:

We expect marker-based tracking to produce rather accurate and steady results under controlled conditions, albeit possibly displaying some jitter during rapid movement, based on general performance characteristics of AR.js for real-time 3D model projection. For fairly complicated models on mid-range devices, rendering performance should be sufficient; nevertheless, frame rates may drop with more polygons or complex textures, especially on less capable hardware. There will probably be a small but visible lag if user interaction is used. The system is anticipated to show that web-based augmented reality is feasible for simple 3D model projection, showcasing AR.js's accessibility while also exposing the performance trade-offs that come with browser-based rendering and tracking in comparison to native applications.

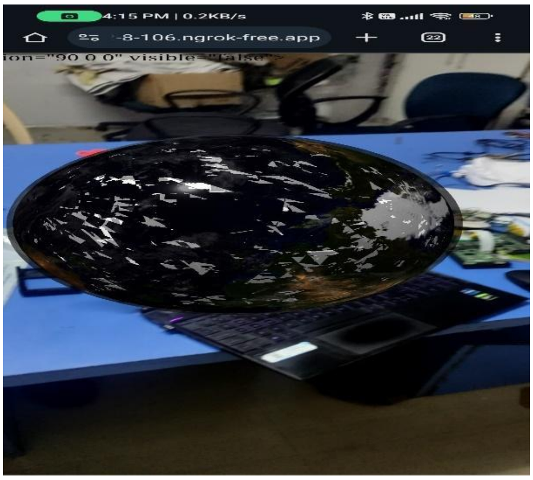

Figure 2.1

Figure 2

Conclusion:

This study effectively used the AR.js package to create and assess a web-based augmented reality system for real-time 3D model projection. The system proved that it could project and, to a certain degree, interact with 3D models in a web browser context without the use of specialized software. The assessment demonstrated AR.js's efficiency for marker-based AR and its advantages for quick prototyping. The performance constraints on devices with limited resources and the difficulties in achieving reliable markerless tracking in web-based settings were also highlighted. According to the results, web-based augmented reality (AR) has a lot to offer in terms of accessibility and platform independence, but 3D model optimization and tracking technique selection should be carefully considered depending on the intended use case and user context. To fully realize the potential of web-based augmented reality, future research should concentrate on discovering more effective rendering approaches, improved markerless tracking algorithms that operate with web browsers, and creating more complex and responsive interaction paradigms.

REFERENCE:

1. Tianrun Chen, Chaotao Ding, Lanyun Zhu, Ying Zang, Yiyi Liao, Zejian Li, "Reality3DSketch: Rapid 3D Modeling of Objects From Single Freehand Sketches", IEEE XPLORE, vol 26, pp 4859 - 4870, 2023.

2. Kun-Chan Lan, Min-Chun Hu, Yi-Zhang Chen, Jun-Xiang Zhang, "The Application of 3D Morphable Model (3DMM) for Real-Time Visualization of Acupoints on a Smartphone", IEEE XPLORE, vol 21, issue:3, pp 3289 - 3300, 2020.

3. Young-Woon Cha, True Price, Zhen Wei, Xinran Lu, Nicholas Rewkowski, Rohan Chabra, "Towards Fully Mobile 3D Face, Body, and Environment Capture Using Only Head-worn Cameras", IEEE XPLORE, vol 24, issue:11, pp 2993 - 3004, 2018.

4. Dan Song, Tian-Bao Li, Wen-Hui Li, Wei-Zhi Nie, Wu Liu, An-An Liu, "Universal Cross-Domain 3D Model Retrieval", IEEE XPLORE, vol 23, pp 2721 - 2731, 2020.

5. Liang Li, Xiuquan Qiao, Qiong Lu, Pei Ren, Ruibin Lin, "Rendering Optimization for Mobile Web 3D Based on Animation Data Separation and On-Demand Loading", IEEE XPLORE, vol 8, pp 88474 - 88486, 2020.

6. Shuai Yu, Xu Chen, Shuai Wang, Lingjun Pu, Di Wu, "An Edge Computing-Based Photo Crowdsourcing Framework for Real-Time 3D Reconstruction", IEEE XPLORE, vol 21, issue:2, pp 421 - 432, 2020.

7. Xiaohui Zhang, Elsa-Marie Otoo, Yubo Fan, Chunjing Tao, Tianmiao Wang, Kawal Rhode, "Autostereoscopic 3D Augmented Reality Navigation for Laparoscopic Surgery: A Preliminary Assessment", IEEE XPLORE, vol 70, issue:4, pp 1413 - 1421, 2022.

8. Maksim Golyadkin, Sergey Saraev, Ilya Makarov, "Advancing Sequential Manga Colorization for AR Through Data Synthesis", IEEE XPLORE, vol 13, pp 7526 - 7537,2025.

9. Onyeka Josephine Nwobodo, Kamil Wereszczyński, Godlove Suila Kuaban, Przemysław Skurowski, Krzysztof Adam Cyran, "An Adaptation of Fitts' Law for Performance Evaluation and Optimization of Augmented Reality (AR) Interfaces", IEEE XPLORE vol 12, pp 169614 - 169627, 2024.

10. Shintaro Imatani, Kensuke Tobitani, Kyo Akabane, "TouchWIM: Object Manipulation in AR Spatial Design With World in Miniature and Hybrid User Interface", IEEE XPLORE, vol 13, pp 69269 - 69280, 2025.