Development of an Immersive Experience for Traditional Media using the Apple Vision Pro

Steffen Büschking, Sarp Güven, Antonin Marxer

***Abstract*—This report proposes an implementation of digital media viewer application as an immersive experience based on the technological novelty of spatial computing introduced with the Apple Vision Pro device, enriching traditional media contents including text and images with digital contents including two-dimensional videos and three-dimensional objects. Despite of the existence of several technical issues, which were mainly caused by software bugs and lacking features in the simulator software of the Apple Vision Pro and the unavailability of the physical device for testing during the development, it was ultimately possible to create a prototype application for viewing digital media content in an immersive environment for the Apple Vision Pro.**

I. Introduction

In the evolving landscape of media consumption, the advent of mixed-reality technologies such as Apple's Vision Pro offers a unique opportunity to address the dichotomy between traditional and digital media. Traditional media's structured narrative and physical engagement contrast sharply with digital media's dynamic interactivity and real-time updates, presenting a gap in delivering a comprehensive media consumption experience. This project introduces a novel digital newspaper application for the Apple Vision Pro, designed to synergize the depth and tactile interaction of traditional media with the immediacy and multimedia capabilities of digital platforms. Our research aims to forge a new paradigm in news consumption by leveraging mixed-reality technology to create an immersive and interactive user experience.

Addressing the integration of diverse media formats within a mixed-reality environment poses significant challenges, including the development of intuitive user interfaces and the seamless incorporation of interactive 3D objects. Our approach, methodologies, and the consequent implementation underscore the potential of mixed-reality applications in revolutionizing how information is consumed. This endeavor not only anticipates the evolving expectations of modern audiences but also sets a benchmark for future developments in mixed-reality content delivery. By documenting the process and outcomes of our project, this report contributes to the body of knowledge in spatial computing, offering insights into overcoming technical hurdles and enhancing user engagement through mixed-reality technologies. Furthermore, this study paves the way for a broader discussion on the implications of mixed-reality applications in other domains, highlighting the versatility and transformative potential of these technologies.

II. Technology Review

The Apple Vision Pro is a mixed reality headset, first introduced at the very end of the Apple Worldwide Developer Conference (WWDC) in June 2023 with a primary focus on consumer rather than companies [2]. It combines capabilities from Augmented Reality (AR) and Virtual Reality (VR) in a single device, although it is technically considered a VR headset with passthrough capabilities supported by two high-resolution main cameras and six world-facing tracking cameras [3]. Apart from that, it consists of a variety of sensors including multiple Light Detection and Ranging (LiDAR) scanners for capturing depth information and four inertial measurement units (IMUs) for sensing positional changes [3]. With these capabilities, the device is able to integrate three-dimensional scenes as well as two-dimensional views into the visible surrounding of the user [1]. On top of that, it enables the creation of fully virtual scenes blended into the passthrough scene for immersive experiences [1]. The blending level between virtual scenes and the passthrough scene is fully adjustable by the user using a mechanical dial [1]. In order to interact with digital content, the Apple Vision Pro supports

skeleton hand tracking using world-facing tracking cameras as well as eye tracking using inside-facing eye tracking cameras [3] [1] [6] [4]. For audio experiences, it features Spatial Audio offering a three-dimensional audio combined with audio effects resulted from raycasting sound waves into the surrounding, considering the physical properties of real-world objects around the user [1]. From the software perspective, the development for Apple Vision Pro requires SwiftUI as a foundational framework, which is capable of interfacing with different Apple-made frameworks like UIKit, ARKit, RealityKit and Metal [6]. Arranging three-dimensional content in the passthrough experience is done via the ARKit framework, implementing several object tracking for the Apple Vision Pro [6]. These object tracking features include World Tracking which is used to persistently store three-dimensional objects in the augmented space, utilizing GPS coordinates and an internally generated map of the surrounding, Plane Detection Tracking for automatically detecting two-dimensional planes in the augmented space based on classifiers (e.g. table or ceiling) and minimum height and width requirements (e.g. square of 10 cm) and Image Tracking recognizing two-dimensional planes containing predefined image patterns [5]. Placing three-dimensional objects into the passthrough environment is achieved using either the RealityKit framework incorporating three-dimensional scenes from Reality Composer Pro, a tool to assemble three-dimensional scenes similiar to Unity and Unreal Engine [6], or the Metal framework, enabling third-party solutions including Unity to provide experiences for the Apple Vision Pro [6] [7].

III. Implementation

For integrating multi-media and three-dimensional content into traditional media, we developed a SwiftUI application capable of displaying different media in the spatial environment of the Apple Vision Pro. The application consist of a two-dimensional main view, which is primarily used for displaying content including text, images and videos, as well as an immersive space, utilized for static and interactive three-dimensional objects and immersive scenes. For compiling the project, we make use of Xcode Version 15.1 beta, embedding the visionOS simulator for testing purposes on the local machine The local machine requires to be an Apple Silicon compatible MacBook or Mac (e.g. MacBook M1 Pro). In order to test the SwiftUI application, we use the official visionOS simulator capable of displaying a virtual environment selectable from a predefined set (e.g. living room, kitchen) instead of the environment captured by the cameras in the Apple Vision Pro in conjunction with the main window of the application. We employed the Reality Composer Pro software embedded into Xcode Version 15.1 Beta, producing object scenes which are imported into the Xcode project and can be directly accessed from the source code in the SwiftUI codebase. The original idea for the application was to project the main view for viewing content on a physical two-dimensional plane. However, we could not pursue that approach due to missing plane detection tracking and image tracking in the visionOS simulator version released with Xcode Version 15.1 Beta and were therefore contrained to implement the majority of the viewer experience in the main window. The application is divided into three parts, which are the windowed application, the immersive space and the content management system.

A. Windowed Application

The main window is meant for displaying the actual content to the user from different media including text, images and embedded previews for purely digital content including videos and three-dimensional objects. The landing page of the application displays a collection of media, from which a specific medium is selected. Each medium is classified by a variety of meta attributes including the author, the available languages and the specific tags by which we can filter the collection. After selecting a medium, its content is displayed across one or multiple pages, through which we can navigate via two buttons in both directions. Each component is constructed from the standard SwiftUI layout components including HStack, VStack etc., divided into multiple files in the component directory. The precise composition of text, images, videos and three-dimensional objects for every page is defined in the MediaState model, injected into the high-level

view, enabling each component of the application to be reactive when the model is updated. Different content types are encoded as nodes in a hierarchical structure and parsed when a medium's content is loaded. In order to define an unified layout for each content type, template files have been specified for text, images, video previews and 3D object previews, which are located in the views directory. The template file for images defines an ImageView for loading images asynchronously, while the template file for videos implements a custom view for loading videos asynchrously and playing them with an instance of the AVPlayer from the AVFoundation framework, which is a cross-platform video framework used by various Apple devices including iPhones, iPads and Mac. In the template file for three-dimensional object previews, we utilized the SCNView class from the SceneKit framework by Apple, allowing to place three-dimensional objects in a two-dimensional Swift View. Each template is rendered as a content block resulting into one or multiple homogeneous sections per page.

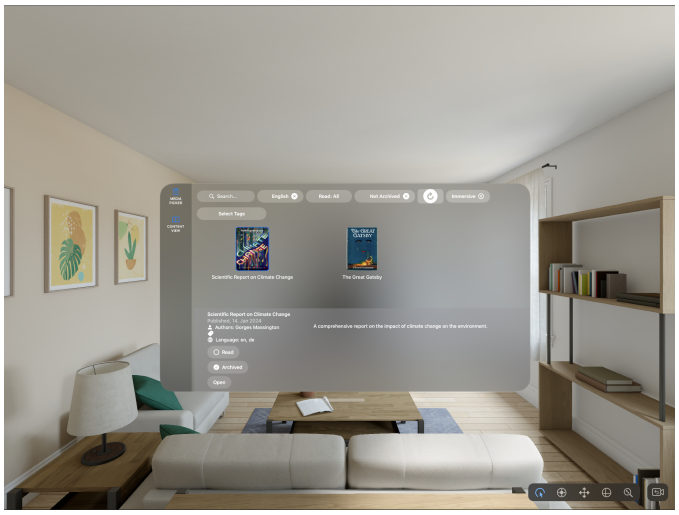

Fig. 1. Landing page of the Media Viewer application

B. Immersive Space

An immersive space defines the world space surrounding the user. It uses the pass-through feature of the Apple Vision Pro. In order to show the capabilities of the immersive space, we developed an interactive earth model, which we created in Reality Composer Pro using a basic sphere model and an earth texture, and then imported into the

Fig. 2. Content View of the Media Viewer application

SwiftUI project. The immersive space was assigned gesture events listening for single tap and double tap, which are used to rotate the earth model by 90 degrees, so that the user can reach every surface point of the earth model. Onto the earth model we placed multiple markers that we designed in Reality Composer Pro using basic sphere and cylinder models. They are set on different locations including Germany, France and Spain, opening location-specific information about the country in a two-dimensional view which is added as a three-dimensional object using an attachment. Additionally, the earth model can be moved freely in the Immersive Space using a dragging gesture. In order to make the movement work, we needed to convert the target coordinates from the SwiftUI coordinate system used by gesture events into the coordinate system used by the immersive space utilizing the given helper functions. Besides the earth model as an interactive object, we implement the inclusion of static three-dimensional objects from a remote source, which is further described in III-C. Furthermore, we included in the main application an immersive scene, which is embedded into the immersive space. It consists of a textured sphere placed at the user's location and surrounding him/her. In order to make the immersion level of the scene adjustable, we change the immersion style from "mixed" to "progressive", allowing the user to freely adjust the level of immersion using the mechanical

dial on the Apple Vision Pro or between three discrete immersion levels including 30%, 50% and 100% in the visionOS simulator.

Fig. 3. Immersive Space including the interactive earth model and the location-based attachment

Fig. 4. Immersive Scene in the background at immersion level 30%

C. Content Management System

The initial plan was to define our media state in a large JSON files, containing all necessary information including meta information of one or multiple media samples as well as the actual content of each media sample. For that purpose, we defined the template system and the MediaState model, decoding the information from the JSON file during the initialization phase. We included an identifier for each content type in order to generate sections of content as described in III-A. Implementing that approach, we faced multiple issues including a higher maintenance effort in terms of addition or deletion of new media samples as well as the absence of content synchronization, adding the requirement of recompilation for any change of the media samples. For that purpose, we decided to deploy a remote Content Management System (CMS) for defining and storing the state related to the media samples of our media viewer application, decoupling the application state from the actual implementation. As an CMS, we deployed a remote Strapi instance [8] and a remote database provided by a custom docker script. The Strapi instance defines the needed collections and their relations among each other. For instance, there is a definition for a medium collection with custom attributes which is related to a page collection defining its own attributes. The content attribute of the page collection enables rich text parsing, supporting text as well as images, videos and three-dimensional objects as hyperlinks referring to the actual image, video or object files. When the content attribute of a page is fetched in the SwiftUI applciation, it is parsed using the template system in order to display the appropriate view (image view, video player etc.) in the appliation. The media state is fetched in the initialization phase of the SwiftUI application. However, reloading the content from the backend system is also made possible via a button in the SwiftUI application, fetching the complete media state from the Strapi system once again.

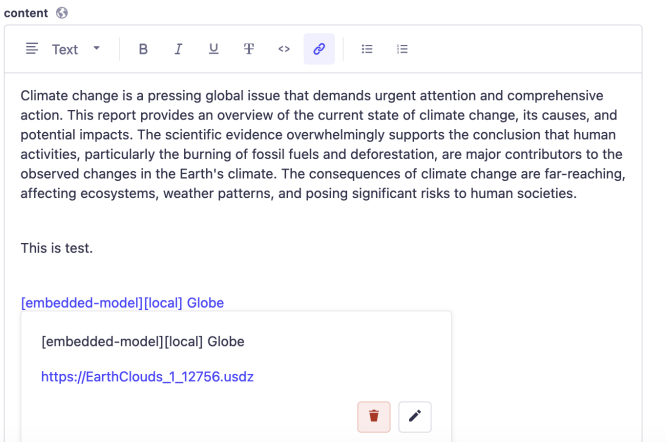

Fig. 5. Strapi Block Editor

IV. Evaluation

In this project, a Media Player was developed for the Apple Vision Pro, marking a significant advancement in the domain of spatial computing. This development integrated conventional media forms, including text and imagery, with digital innovations such as interactive three-dimensional objects and immersive environments. The project objectives, delineated in the initial workshop, were realized in two phases, each executed without encountering substantive impediments. Furthermore, a content management system was devised and deployed on proprietary servers, facilitating the customization of media content, albeit with the limitation of adjusting interactive 3D objects.

Notwithstanding these achievements, the project confronted notable challenges. The ambition to project media onto physical surfaces was curtailed due to the lack of tracking capabilities in the visionOS Simulator Beta. This limitation necessitated compromises, notably in the realm of 3D sound experiences and the integration of 3D objects through Reality Composer Pro, attributed to extant bugs within the visionOS simulator. Additional problems were encountered with the integration of 3D content, particularly due to corrupted USDZ file formats, which precluded their export to the backend, necessitating local loading. Conversion issues between RealityKit and SwiftUI further complicated the development process, disrupting the seamless integration of mixed-reality content.

Complications were further compounded by difficulties with visionOS-specific functionalities, such as the reactivity in immersive spaces and the conversion accuracy between SwiftUI and RealityView units. These difficulties were magnified by the scarcity of comprehensive documentation and the complexity inherent in available sample projects. Additionally, the despawning of objects in immersive spaces posed a significant challenge. To mitigate this, a workaround was implemented, enabling the reloading of the complete scene for any changes to be effected, thus ensuring continuity in the immersive experience.

With the commercial release of the Apple Vision Pro, it is anticipated that the incidence of simulator-related issues will diminish. It is expected that the device's actual deployment will facilitate enhanced testing of tracking and sensor functionalities. Moreover, the expansion of the developer community is likely to catalyze the availability of more detailed documentation and tutorials, thereby mitigating many of the technical challenges previously encountered.

The endeavor to develop a digital newspaper application for the Apple Vision Pro has been characterized by a continuous cycle of innovation, learning, and adaptation. Despite facing several technical challenges, substantial progress has been made towards delivering an immersive and interactive news consumption experience. This project's journey underscores a commitment to refining the application and surmounting obstacles, with a view towards redefining the paradigms of news consumption within a mixed-reality context. The insights gleaned from this project contribute valuably to the corpus of knowledge in spatial computing, underscoring the transformative potential of mixed-reality technologies in reimagining conventional media consumption practices.

V. Conclusion

In this report, we could show that it is possible to develop a media player application utilizing the Apple Vision Pro's capabilities to combine fully digital content including videos and three-dimensional objects with traditional content including text and images, which can be fetched from a centralized remote server. These new capabilities of the Apple Vision Pro open the path to new user interaction types and new behaviours on how to consume media, especially considering the entertainment domain due to the immersive state achievable with our proposed implementation. Beyond entertainment, this project highlights the potential impact on the scientific domain as users may gain better understanding of objects and their behaviours when they are displayed in a three-dimensional view, allowing the user to make observations from all angles. Moreover, we added the capability of attaching additional information to objects, introducing a distinct form of representation in the three-dimensional world. Although we were not able to achieve a fully immersive experience due to the lack of tracking features in the simulator application in the context of

this project, we are certain that it will be possible to implement such an application capable of projecting media content to a physical object with the help of tracking features included in the Apple Vision Pro in the future.

REFERENCES

[1] Apple. *Introducing Apple Vision Pro*. June 2023. URL: https://www.youtube.com/watch?v=TX9qSaGXFyg (visited on 02/27/2024).

[2] Apple. *WWDC 2023 — June 5 — Apple*. June 2023. URL: https://www.youtube.com/watch?v=GYkq9Rgoj8E (visited on 02/27/2024).

[3] *Apple Vision Pro - Technical Specifications*. en-US. URL: https://www.apple.com/apple-vision-pro/specs/ (visited on 02/27/2024).

[4] *ARKit in visionOS*. en-US. URL: https://developer.apple.com/documentation/arkit/arkit_in_visionos (visited on 02/27/2024).

[5] Apple Inc. *Meet ARKit for spatial computing - WWDC23 - Videos*. en. URL: https://developer.apple.com/videos/play/wwdc2023/10082/ (visited on 02/27/2024).

[6] Apple Inc. *visionOS Overview*. en. URL: https://developer.apple.com/visionos/ (visited on 02/27/2024).

[7] Skarredghost. *First look at Apple Vision Pro development in Unity*. en-US. Nov. 2023. URL: https://skarredghost.com/2023/11/24/vision-pro-development-unity/ (visited on 02/27/2024).

[8] *Strapi's documentation — Strapi Documentation*. en. URL: https://docs.strapi.io/ (visited on 02/28/2024).