Demo: Real-Time WebXR Edge-based Object Detection for AR

Jacky Cao
jacky.cao@oulu.fi
University of Oulu
Oulu, Finland

Kit Yung Lam
kylambd@connect.ust.hk
Hong Kong University of Science and Technology
Hong Kong

Lik-Hang Lee
lik-hang.lee@polyu.edu.hk
Hong Kong Polytechnic University
Hong Kong

ABSTRACT

Web-based extended reality (WebXR) can enable lightweight, easy-to-access, and cross-platform augmented reality (AR) experiences. Context awareness is one key feature of AR. Supporting this in browser-based WebXR applications is challenging as typical object detection algorithms are too computationally demanding to be run in-browser, leading to slow response times and decreased battery life. In this demo, we show a WebXR AR application that uses a technique of WebRTC-based video streaming to obtain a usable video stream on an edge server to perform object detection.

CCS CONCEPTS

• **Human-centered computing** → **Mixed / augmented reality**;
• **Information systems** → **Web applications**.

KEYWORDS

augmented reality, webxr, edge computing

ACM Reference Format:
Jacky Cao, Kit Yung Lam, and Lik-Hang Lee. 2023. Demo: Real-Time WebXR Edge-based Object Detection for AR. In *The 21st Annual International Conference on Mobile Systems, Applications and Services (MobiSys '23), June 18–22, 2023, Helsinki, Finland.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3581791.3597286

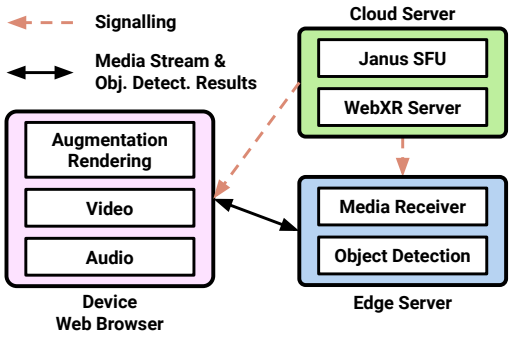

Figure 1: The pipeline of the AR system.

1 INTRODUCTION

With increasing interest in the Metaverse [1] and the ongoing popularity of augmented reality (AR), achieving large-scale public adoption requires ubiquitous access to applications that provide a high quality of service and quality of experience (QoE), which highly relies on understanding user context [9]. Thus, AR applications utilise onboard sensors of user devices such as optical cameras to capture environmental data for analysis, which then supports context-relevant augmentations shown to users. Nonetheless, this analysis is often offloaded to external edge [7, 8] or cloud servers [10] as the user devices lack the computational capability to do so in sufficient time. At the same time, these applications are traditionally install-based, i.e., from an application store, which could limit the willingness of users as they have to download and then subsequently install a proprietary application that provides just one AR experience. Additionally, from the developers' perspective, if AR applications are to be developed natively, then this would mean developing and recompiling the applications for several smartphone operating systems, which could take significant effort and work. Considering the device hardware itself, while users can procure specialised hardware to access AR experiences, such as head-mount displays (HMDs), they are more likely to have a smartphone already.

With these constraints in mind, namely, developing cross-platform AR applications that are usable on a variety of heterogeneous smartphone hardware, one way to address this is to utilise web-based applications, i.e., WebXR (Web-based Extended Reality) or WebAR (Web-based AR) [6], where WebXR is a continually developed group of web standards for rendering augmentations on the real-world (AR) or virtualising entire 3D environments (virtual reality) [4]. These applications are deployed on external servers and directly accessed in user device web browsers. This offers several benefits, such as platform-independent applications, ease of access, scalability, and interoperability. However, achieving performance parity with traditional install-based applications is challenging, especially in supporting real-time object detection. This feature is essential for context awareness and delivering relevant augmentations to users in AR applications. While browsers can run native JavaScript libraries such as TensorFlow.js [3] and MediaPipe [2] to execute object detection tasks in-browser, this is at the expense of increased device energy usage to perform the calculations.

Therefore, a viable solution is exploiting WebRTC Selective Forwarding Unit (SFU) used for peer-to-peer video and audio streaming in WebXR-based AR applications. This demo presents an AR system that performs seamless object detection and is deployable on the edge and cloud. This system can be easily adapted to accommodate different application scenarios, e.g., education, manufacturing, etc., through changing the available content rendered to users.

MobiSys '23, June 18–22, 2023, Helsinki, Finland

ACM ISBN 979-8-4007-0110-8/23/06.
https://doi.org/10.1145/3581791.3597286

2 SYSTEM OVERVIEW

Figure 1 shows the AR system pipeline containing components for device web browsers, an edge server, and a cloud server. The pipeline is as follows: 1) a user loads the AR application in their browser; 2) the WebXR server receives the request, and the application is sent to the browser; 3) simultaneously, the Janus SFU begins WebRTC signalling to establish that the user's browser is available to share the smartphone's video and audio streams; 4) the edge server is pre-connected with the Janus SFU and is notified that new media data is available; 5) the edge and browser-loaded application perform handshaking through the cloud server; 6) the edge receives the smartphone's media streams and performs object detection; and 7) results are returned to the smartphone and rendered.

3 DEMONSTRATION

To illustrate the WebXR system, an edge, a cloud, and user devices access and test the system. Their specifications are: the edge is a PC with an Intel Core i7-9750H CPU, 32 GB RAM, and an NVIDIA GeForce RTX 2080 GPU; the cloud is an Amazon EC2 gd4n.2xlarge cloud server with 8 vCPUs, 32 GB RAM, and an NVIDIA T4 GPU. Then, a OnePlus 9 Pro, iPhone 13 Pro Max, and a Microsoft HoloLens 2 are the clients. These clients and the edge server are connected to a 5G Test Network at the University of Oulu, Finland.

The WebXR framework, Networked A-Frame [5], is the backbone for the prototype AR application as the framework supports WebRTC connections for media and data communication, is cross-platform, and is highly extensible. This allows us to deploy the novel object detection plugin and simultaneously support collaborative AR experiences. This framework is deployed on the cloud server and co-hosted with a Janus WebRTC server to manage the signalling between clients and the edge server. A custom service is deployed on the edge server to interface with the Janus server, which retrieves video and audio streams from user devices and then performs YOLOv8-based object detection on the video streams.

The user client is configured to show the recognised objects' labels from the object detection service in this demonstration. Figure 2 shows an example view of the AR labelling when the application is loaded in-browser on a smartphone. Figure 3 shows the corresponding view when the application is loaded in-browser on a HoloLens 2. Modern smartphones can render the labels in near real-time, whereas with the HoloLens 2, the labels momentarily lag due to resource constraints on the mobile headset. However, the WebXR system can still analyse the user's surrounding environments, regardless of the rate at which the labels are rendered.

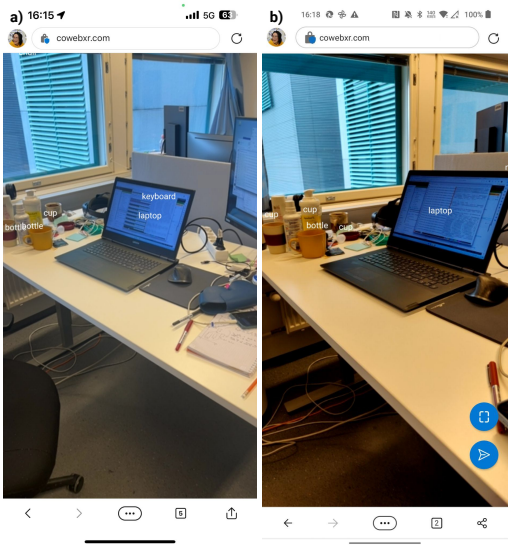

Figure 2: Example views of the AR application in-browser on smartphones, using Microsoft Edge, the application loaded on: a) an iPhone 13 Pro Max and b) a OnePlus 9 Pro. The user sees a passthrough video stream from their device's world-facing camera, on that stream are labels returned from the object detection service and rendered on top of the stream.

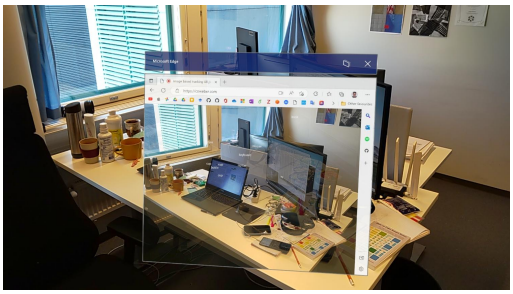

Figure 3: Example view of the AR application in Microsoft Edge on a HoloLens 2. The browser hologram shows a passthrough stream from the world-facing camera and rendered object labels from the object detection service.

REFERENCES

[1] Lik-Hang Lee et al. 2022. What is the Metaverse? An Immersive Cyberspace and Open Challenges. *ArXiv* abs/2206.03018 (2022).
[2] Google. 2020. MediaPipe | Live ML anywhere. https://mediapipe.dev/
[3] Google. 2023. TensorFlow.js. https://www.tensorflow.org/js
[4] Mozilla. 2020. WebXR Device API | Web APIs. https://developer.mozilla.org/en-US/docs/Web/API/WebXR_Device_API
[5] Networked-Aframe. 2023. Networked-aframe. https://github.com/networked-aframe/networked-aframe
[6] Xiuquan Qiao, Pei Ren, Schahram Dustdar, Ling Liu, Huadong Ma, and Junliang Chen. 2019. Web AR: A Promising Future for Mobile Augmented Reality—State of the Art, Challenges, and Insights. *Proc. IEEE* 107, 4 (2019), 651–666. https://doi.org/10.1109/JPROC.2019.2895105
[7] Jovan Stojkovic, Zida Liu, Guohao Lan, Carlee Joe-Wong, and Maria Gorlatova. 2019. Edge-Assisted Collaborative Image Recognition for Augmented Reality: Demo Abstract. In *Proceedings of the 17th Conference on Embedded Networked Sensor Systems* (New York, New York) *(SenSys '19)*. Association for Computing Machinery, New York, NY, USA, 394–395. https://doi.org/10.1145/3356250.3361944
[8] Xiang Su, Jacky Cao, and Pan Hui. 2020. 5G Edge Enhanced Mobile Augmented Reality. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking* (London, United Kingdom) *(MobiCom '20)*. Association for Computing Machinery, New York, NY, USA, Article 64, 3 pages. https://doi.org/10.1145/3372224.3417315
[9] Wenxiao Zhang et al. 2022. EdgeXAR: A 6-DoF Camera Multi-Target Interaction Framework for MAR with User-Friendly Latency Compensation. *Proc. ACM Hum.-Comput. Interact.* 6, EICS, Article 152 (jun 2022), 24 pages. https://doi.org/10.1145/3532202
[10] Zhuo Zhang, Pan Hui, Sanjeev Kulkarni, and Christoph Peylo. 2014. Enabling an Augmented Reality Ecosystem: A Content-Oriented Survey. In *Proceedings of the 2014 Workshop on Mobile Augmented Reality and Robotic Technology-Based Systems* (Bretton Woods, New Hampshire, USA) *(MARS '14)*. Association for Computing Machinery, New York, NY, USA, 41–46. https://doi.org/10.1145/2609829.2609835