Edge-Centric Augmented Reality Framework for Realtime Wristwatch Try-On

Stevica Cvetković*, Matija Špeletić, Jelena Nikolić

University of Niš, Faculty of Electronic Engineering, 18000 Niš (Serbia)

* Corresponding author: stevica.cvetkovic@elfak.ni.ac.rs

Received 1 July 2025 | Accepted 28 January 2026 | Early Access 14 May 2026

Abstract

The rapid expansion of online retail has intensified the need for realistic and interactive product visualization. Virtual try-on technologies have emerged as a critical tool for enhancing user confidence and reducing product return rates. Most existing research has focused on apparel-oriented solutions that rely on computationally intensive algorithms. In contrast, comparatively little attention has been given to smaller accessories such as wristwatches and jewelry, which present unique modeling challenges due to their scale and placement. Furthermore, the deployment of such systems on resource-constrained edge devices remains largely underexplored. In this work, we present a markerless augmented reality framework for wristwatch try-on, optimized for execution on smartphones and web browsers to enable real-time, privacy-preserving operation without reliance on cloud processing. The framework incorporates hand pose estimation, local 3D rendering, and buffer-based geometric parameter smoothing. Our approach integrates a hand landmark detection algorithm capable of estimating the watch model's 3D position from three key hand landmarks, and introduces a buffer-based method for smoothing geometric parameters during movement. Features such as photorealistic reflections and physics-based materials are outside the current modeling scope. Our primary contribution is a lightweight, edge-executable pipeline for small-accessory try-on that achieves interactive frame rates (>30 fps) and a high level of visual quality. Evaluations using smartphone and web cameras demonstrate competitive rendering stability, with a mean opinion score of 4.35 on an introduced dataset, indicating that the augmented frames were generally perceived as highly realistic. These results demonstrate the feasibility of delivering immersive AR try-on for small accessories on edge-devices, offering a viable alternative to cloud-based solutions in online retail.

Keywords

Augmented Reality, Edge Computing, Hand Pose Estimation, Human-Computer Interaction (HCI), Virtual Try-On.

DOI: 10.9781/ijimai.2026.2227

I. Introduction

THE growing demand for personalized e-commerce experiences has amplified the importance of realistic and interactive product visualization, positioning virtual try-on technologies as a key driver of consumer confidence and reduced return rates. The advent of virtual try-on systems has introduced a new era of consumer engagement and convenience within the e-commerce landscape [1]. The potential of augmented reality (AR) and artificial intelligence (AI) technologies to transform the way customers explore and purchase fashion items is increasingly evident [2]. AR systems are designed to enhance the interaction between real and virtual objects by integrating them seamlessly within real-time environments. These systems must effectively blend virtual objects with the physical world while ensuring that interactions occur with minimal latency, thereby facilitating a more immersive user experience [3]. Virtual Try-On (VTO) systems, with their capacity to digitally simulate the fitting experience, have been instrumental in mitigating the challenges associated with online shopping, including the inability to assess the physical compatibility of products with customer's body [4].

Building on recent advances in AR-based virtual try-on applications for clothing, this paper addresses the gap in watch retail by focusing on efficient, real-time, and privacy-preserving execution on edge devices. Watches, as prominent fashion accessories, present unique considerations in consumer evaluation. To address these, we propose a novel virtual try-on system that allows users to visualize, interact with, and assess the compatibility of different watch models on their wrists in a virtual environment. This approach is designed to enhance customer confidence, improve satisfaction, and reduce return rates, thereby supporting growth in the online wristwatch market. Our research contributes a markerless AR framework for virtual wristwatch try-ons, emphasizing usability by removing the need for markers and ensuring efficient real-time execution on edge devices.

Please cite this article as:
S. Cvetković, Matija Špeletić, Jelena Nikolić. Edge-Centric Augmented Reality Framework for Realtime Wristwatch Try-On, *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 9, no. 7, pp. 78-87, 2026, http://doi.org/10.9781/ijimai.2026.2227

In this paper, we present the architectural design and development of our markerless VTO system, focusing on its edge-based execution. We analyze the system's key components and algorithms and evaluate its performance and potential implications for real-world applications. The proposed framework exploits edge-based execution [5], eliminating the need for server-side processing and ensuring privacy, efficiency, and real-time performance. The framework comprises several key components. First, it performs accurate estimation of hand landmark positions in 3D space using a robust on-device hand landmark detection algorithm. Subsequently, an efficient tracking algorithm is applied to maintain performance in consecutive frames. To achieve realistic virtual 3D model rendering with a correct viewpoint, the overlays of the target wristwatch model must appear in precise alignment with the user's hand. We compute the scale, position, and rotation of the 3D watch model based on the detected hand landmarks through a series of real-time calculations executed entirely on the edge device. To achieve a smooth effect when displaying the 3D watch model over the moving hand, we compute a series of buffers to eliminate jitter and improve user experience. The buffered scale, position and rotation parameters are then used to correctly display the watch model on a transparent canvas placed over the live camera stream. Performance evaluations in web browsers and smartphones confirmed the system's ability to deliver a seamless virtual try-on experience with real-time processing speeds. By exploiting the intersection of artificial intelligence, augmented reality, and edge computing, our work advances VTO technology and contributes to the ongoing evolution of e-commerce platforms.

Our main contribution is a real-time, edge-based augmented reality framework for wristwatch virtual try-on, which integrates accurate 3D hand landmark detection, stable geometric augmentation, and a practical web-based implementation, validated through empirical testing on consumer devices. The specific contributions of the paper are presented below:

1. Novel framework for wristwatch virtual try-on: The paper introduces a real-time augmented reality framework specifically designed for wristwatch virtual try-on, addressing a gap in the literature dominated by apparel-based solutions. It incorporates a robust hand landmark detection algorithm to accurately estimate the 3D position of the watch model based on three distinct hand landmarks. A buffer-based approach for geometric parameter processing ensures stable and smooth 3D augmentation during dynamic hand movements.
2. Edge-based processing: All computational tasks are executed entirely on edge devices, including demanding operations such as hand pose estimation, 3D rendering, and parameter buffering. This edge-centric design ensures efficient, real-time performance while maintaining user privacy by eliminating the need for server-side computation.
3. Practical implementation and evaluation: The framework is implemented as a functional prototype in the form of a web-based application, featuring a generic architecture that can serve as a baseline for other types of AR web applications. Empirical evaluations conducted using smartphones and web cameras demonstrate the system's capability to deliver high visual quality (MOS = 4.35), real-time responsiveness (>30 fps), and practical feasibility for online retail applications.

The remainder of this paper is organized as follows: Section II provides an overview of related work in virtual try-on systems, Section III delves into the technical design and implementation of our VTO system, while Section IV presents the evaluation results. Section V considers the implications of our findings, and Section VI concludes the paper.

II. Related Work

VTO systems enhance the online shopping experience by simulating real-world product interactions. As Scholz and Duffy [4] highlighted, VTO can strengthen consumer-brand connections by integrating product experiences into users' personal spaces, thus fostering engagement and trust. Song et al. [16] found that AR try-on increases decision comfort by enhancing immersion and the feeling of ownership, leading to more confident choices. The high interactivity and realism of VTO address a key limitation of e-commerce: the inability to physically assess products. Kim et al. [17] underlined that AR and VR try-on systems elevate sensory engagement, providing a sense of presence that boosts purchase intention. Arya et al. [18] emphasized how brands utilize virtual experiences and gamification in the Metaverse to foster loyalty. According to Kim and Forsythe [1], these immersive technologies reduce product risk and improve shopping enjoyment, promoting the adoption of VTO in consumer decision-making. Furthermore, Galán et al. [19] demonstrated that the medium of presentation, such as virtual reality or augmented reality, significantly impacts user perception of products. Given its significant impact, VTO is becoming essential in e-commerce, transforming consumer interactions with products.

Several studies have explored image-based virtual try-on methods that do not utilize 3D information. For instance, Han et al. proposed the VITON (Virtual Try-On Network) model [6], which employs a two-step refinement process to overlay garments on a person in the same pose while preserving visual detail. Unlike methods requiring complex spatial transformations, VITON uses a clothing-agnostic person representation to create a seamless fitting experience. Similarly, Wang et al. introduced CP-VTON [7], which retains clothing identity details, such as texture and logos, by integrating a Geometric Matching Module (GMM) for spatial alignment and a Try-On Module to enhance clothing integration. These models improve image-based try-on by addressing spatial misalignment, a common issue in traditional 2D approaches. Building on this, Minar et al. [8] developed CP-VTON+, which optimizes clothing representation to address alignment distortions and composition artifacts, thereby enhancing realism for challenging body poses. Chong and Mo [9] utilized vision transformers in ST-VTON, adopting a self-supervised approach to train on unpaired datasets, which enhances model versatility across clothing and pose variations without labeled data. In contrast, 3D-based methods allow for more intricate interactions between clothing and human models. For example, Duan et al. [10] proposed a method based on three-dimensional garment scans and virtual tailoring. Their approach decomposes both the garment and human model into patches aligned through feature matching, followed by virtual sewing. This 3D-focused system enables detailed garment fitting by preserving geometrical integrity, making it particularly effective for items requiring realistic draping and contouring. There is also significant potential for incorporating multimodal sensor fusion, empowered by advanced deep learning models, to enhance pose estimation and user interaction. Recent studies on point-cloud-based hand gesture recognition [20] and human motion behavior recognition [21], demonstrate promising directions.

Wristwatch virtual try-on has also gained popularity, particularly through AR implementations, which can either be marker-based or markerless. Wu et al. [11] introduced a system that enables real-size virtual try-on for watches, requiring preprocessing step of camera calibration to maintain accurate size perception. This system uses 3D virtual models, providing an immersive experience through coordinate transformations compatible with modern web frameworks. Mohan [12] focused specifically on AR for watches on mobile devices using a marker-based approach. In this application, users scan a

TABLE I. Comparison of Related Papers With the Proposed Framework

	Ours	VITON [6]	CP-VTON [7]	CP-VTON+ [8]	ST-VTON [9]	Garment Fit [10]	Real-Size AR [11]	AR Watch [12]	ARZARA [13]	GlamTry [14]	HCAN [15]
Markerless Detection	Yes	Yes	Yes	Yes	Yes	No (3D scanner)	Yes	N/A	No (marker band)	Yes	Yes
Result Preview	3D	2D	2D	2D	2D	3D	3D	3D	3D	2D	2D
Real-time execution	Yes	No	No	No	No	No	Yes	Yes	Yes	No	No
On-device processing	Yes	No	No	No	No	No	Yes	Yes	Yes	Yes	No
Supported devices	Mobile/ Web	PC	PC	PC	PC	PC	Mobile/ Web	Mobile	Mobile	Mobile/ Web	PC

marker with their device to see a virtual watch overlaid on their wrist. Mohan's marker-based AR implementation offers users an engaging, life-like experience by providing interactivity through AR. Likewise, Gupta et al. [13] developed ARZARA, a Unity-based AR app that uses marker-based technology to display watches on a user's wrist. This application aims to improve the online watch shopping experience by allowing users to visualize various watch models and sizes, helping reduce the need for physical trials. A virtual try-on method for accessories such as jewelry and watches was proposed by Chang and Lekena [14], adapting 2D clothing try-on techniques. This approach serves as a bridge between garment-focused and accessory-specific applications. However, it yields limited visual quality and has only been evaluated on static images. An image-based virtual try-on system presented by Tang et al. [15] leverages state-of-the-art hierarchical cross-attention networks to enhance the realism of clothing overlays. Although developed for garments, its architectural innovations hold potential for cross-domain adaptation. Nevertheless, its relatively high computational complexity constrains deployment on edge devices.

Building on insights from previous studies, our research focuses on developing a markerless virtual try-on system for wristwatches, eliminating the need for preprocessing steps such as camera calibration. The proposed framework enables markerless detection and tracking of the wrist and hand from a standard RGB video stream. Furthermore, it is designed to operate seamlessly within a web browser, removing the need for specialized software installation. Running in real time on-device, it provides an interactive 3D wristwatch preview, allowing users to try watches directly in a browser or mobile application. Table I compares the proposed framework with related works. Although the Real-Size AR [11] approach achieves similar functionality, our method improves landmark stability and computational efficiency through temporal smoothing and a simplified processing pipeline, enhancing robustness on resource-constrained devices.

III. Novel Virtual Try-On Framework

We use built-in web cameras or smartphone cameras to acquire a live video feed, which is processed entirely on edge devices using JavaScript technology. A simplified workflow diagram of the proposed AR system is shown in Fig. 1. Each video frame is handled locally without any server-side computation, ensuring fast, efficient, and privacy-preserving processing. Initially, the frame is passed to the hand landmark detection model [22], which operates on-device to detect and track hand landmarks in real-time. If a hand was detected in the previous frame, a lightweight tracking algorithm is employed to reduce computational overhead. Otherwise, the frame is processed by the on-device hand landmark detector. Using the estimated landmarks, our framework computes the scale, position, and rotation of the 3D watch model through a series of on-device calculations. To ensure smooth and stable visualization of the 3D watch model during hand movement, we implement a series of buffers that eliminate extreme values and interpolate new ones as needed. These buffered parameters are then applied to accurately render the watch model on a transparent canvas layered over the live camera feed. The entire pipeline, from video acquisition to augmented reality visualization, is executed within a web browser, eliminating the need for server communication and enabling real-time responsiveness.

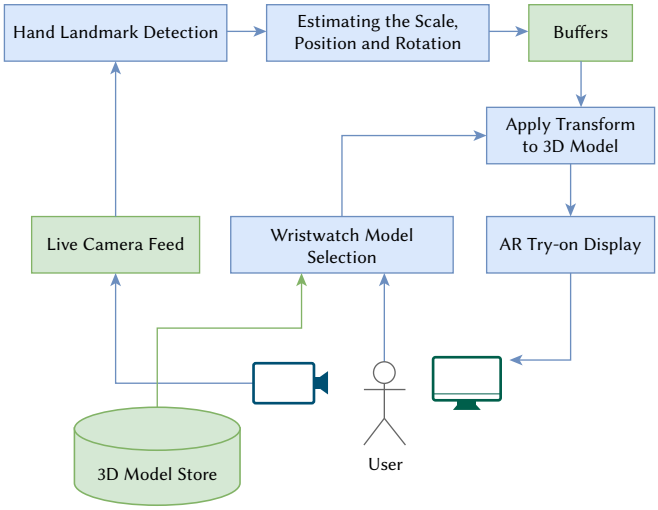

Fig. 1. Architecture of the proposed AR-based wristwatch try-on framework.

While numerous methods exist for highly accurate hand detection [23], [24], [25], [26], our objective is to ensure that the entire process is sufficiently optimized for execution on resource-constrained devices. Therefore, we are utilizing a hardware accelerated deep learning model from the MediaPipe framework [27], which supports efficient AI inference directly on edge devices [22]. The rendering and display of the models are handled by the Three.js library [28]. The system processes the live camera feed and the user's selected watch model entirely on-device, delivering an AR try-on experience in real-time. The server-side component consists solely of a database that stores 3D models in GLTF format along with corresponding metadata and thumbnails. The detailed architecture of our framework is shown in Fig. 2. All processing required for real-time operation is performed locally on the client device, leveraging convenient edge-based computing [5], [29]. In the following subsections, we describe each step of the framework in detail.

A. On-Device Hand Landmark Detection

An edge-based model for hand landmark detection and tracking is crucial for the proposed system, as it ensures low-latency processing

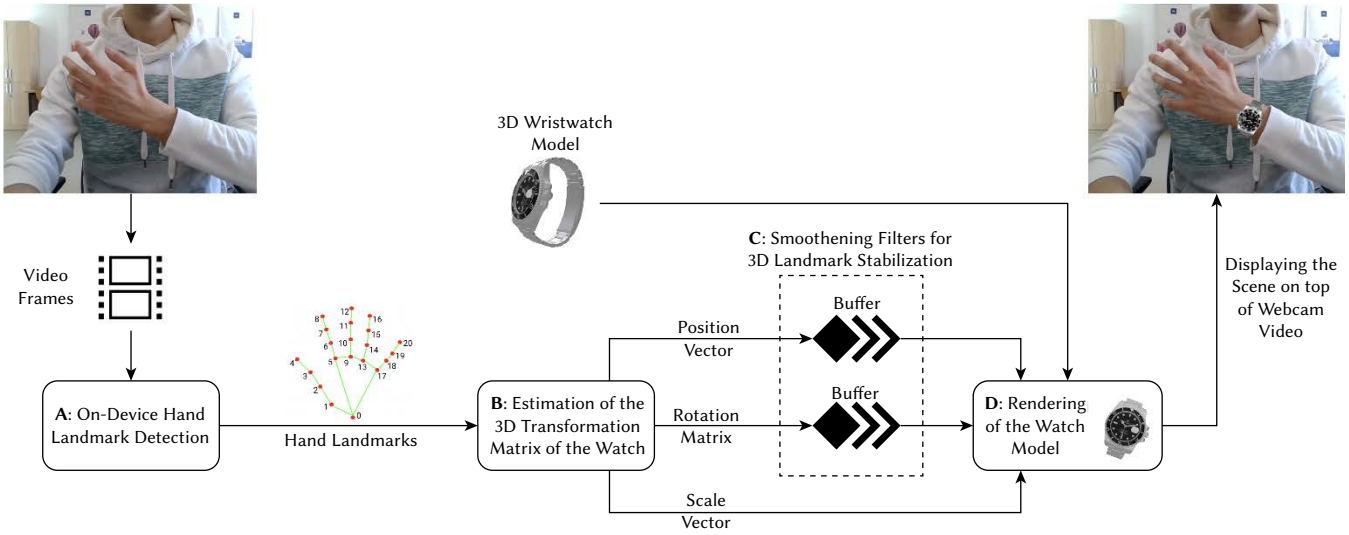

Fig. 2. Workflow diagram of our augmented reality wristwatch try-on framework. First, the webcam frames are passed through the hand landmarks detection model (A). These landmarks are used to estimate the watch's 3D transformation matrix (B). Position and rotation values are smoothed using an exponential window filter (C). The 3D watch model is updated with the smoothed motion. It is then rendered and overlaid on the live video (D).

and enhances the feasibility of deploying the system on mobile and resource-constrained devices. By performing computations closer to the data source, edge-based models reduce reliance on cloud infrastructure, ensuring faster inference times and minimizing privacy concerns related to data transmission. This approach aligns with previous research demonstrating the effectiveness of edge computing in achieving real-time performance while addressing user concerns about data privacy [30].

The MediaPipe hand tracking and landmark detection pipeline [22] consists of a palm detection model and a hand landmarks detection model. The palm detection model locates hands within the input image, and the hand landmarks detector identifies specific hand landmarks on the cropped hand image defined by the palm detector. In video mode, the hand landmarker uses the bounding box defined by the hand landmarks model in one frame to localize the region of hands for subsequent frames. It re-triggers the palm detection model only if the hand landmarks model no longer identifies the presence of hands or fails to track the hands within the frame. Fig. 3 shows the hand model, defined by 21 landmarks, as returned by the MediaPipe hand landmarker. To estimate the position, scale, and rotation of the hand (i.e., the watch model), we rely on three representative hand landmarks: WRIST, INDEX_MCP, and PINKY_MCP – located at indexes 0, 5, and 17, respectively in the 21-joint MediaPipe hand model (Fig. 3). Our method links the position, rotation, and scale of the watch model in 3D space to those of the hand.

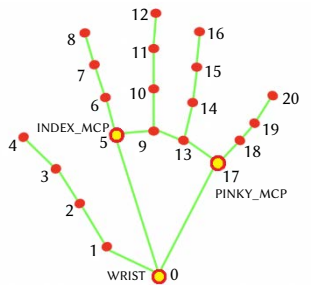

Fig 3. The 21-joint hand model from the MediaPipe hand landmarker [22], where our method focuses on three landmarks - WRIST (0), INDEX_MCP (5), and PINKY_MCP (17).

B. Estimation of the 3D Transformation Matrix of the Watch

To correctly place the 3D watch model in the scene we have to estimate the transformation M defined by the following 4×4 homogenous matrix:

$$\boldsymbol{M} = \boldsymbol{TRS} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \boldsymbol{R} \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where $\boldsymbol{t} = [t_x \quad t_y \quad t_z]^T$ is the position vector, $\boldsymbol{R}$ is the rotation matrix, and $\boldsymbol{s} = [s_x \quad s_y \quad s_z]$ the scale vector [31].

1. Position Vector Estimation

The positioning of the watch model is primarily determined by the location of the wrist. A straightforward approach would be to render the watch model directly over the wrist. However, considering that a wristwatch typically rests slightly away from the wrist on the arm, we propose a method to more accurately estimate the watch's position. This method utilizes the INDEX_MCP, PINKY_MCP and WRIST landmarks, which are represented by the $\boldsymbol{p}_{index}$, $\boldsymbol{p}_{pinky}$ and $\boldsymbol{p}_{wrist}$ vectors, respectively, as illustrated in Fig. 4.

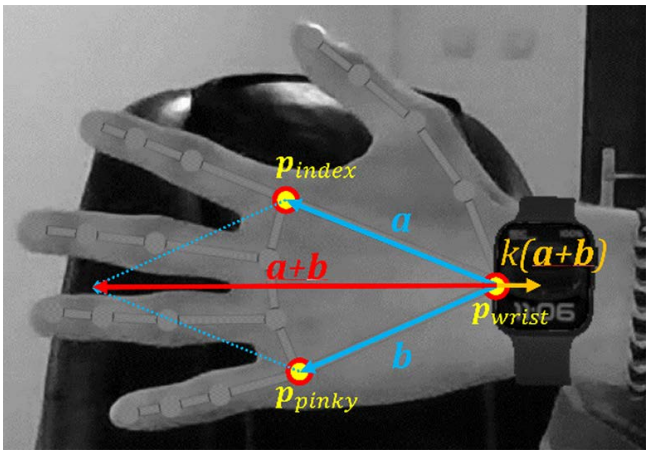

Fig. 4. The two vectors, a and b, were used to estimate the watch model position for rendering. The INDEX_MCP, PINKY_MCP and WRIST landmarks are represented by the $\boldsymbol{p}_{index}$, $\boldsymbol{p}_{pinky}$ and $\boldsymbol{p}_{wrist}$ vectors.

The position vector (t) of the wristwatch is calculated using the following formula:

$$\begin{aligned} \boldsymbol{t} &= k_t(\boldsymbol{a}+\boldsymbol{b}) \\ \boldsymbol{t} &= k_t\left((\boldsymbol{p}_{index}-\boldsymbol{p}_{wrist})+(\boldsymbol{p}_{pinky}-\boldsymbol{p}_{wrist})\right) \\ \boldsymbol{t} &= k_t(\boldsymbol{p}_{index}+\boldsymbol{p}_{pinky}-2\boldsymbol{p}_{wrist}) \end{aligned} \tag{2}$$

where k_t is a constant determining the distance of the watch model from the wrist. For the constant k_t a small negative value is empirically determined to place the watch slightly away from the wrist and towards the elbow.

The calculated vector t, should be added to the position vector of the WRIST point in order to obtain the final position of the wristwatch. Additionally, we make $t_z = 0$ to ensure proper clipping, which is explained in detail in Section III.D.

2. Scale Vector Estimation

We can assume that the watch model is scaled uniformly along all axes, so that the components of the scale matrix are identical:

$$s_x = s_y = s_z = s \tag{3}$$

This uniform scaling approach ensures consistent visual representation of the watch model relative to the detected hand size. To estimate the scale factor for the watch model, we calculate the circumference of the triangle formed by three hand landmarks: INDEX_MCP, PINKY_MCP, and WRIST. This value is then multiplied by an empirically determined constant to scale the watch model according to hand size. The scale factor s is computed as follows:

$$s = k_s(\|\boldsymbol{p}_{index}-\boldsymbol{p}_{wrist}\|+\|\boldsymbol{p}_{pinky}-\boldsymbol{p}_{wrist}\|+\|\boldsymbol{p}_{pinky}-\boldsymbol{p}_{index}\|) \tag{4}$$

where $\|x\|$ indicates the Euclidean norm of vector $\boldsymbol{x}$.

3. Rotation Matrix Estimation

Estimating the 3D rotation matrix for the watch model represents the most complex step in the process. To derive the 3D rotation matrix based on three specified points in space, we utilize the following consideration. The rows of a 3×3 rotation matrix (the first three rows and columns of a 4×4 homogeneous rotation matrix) can be interpreted as the unit vectors that define the rotated 3D space. This interpretation is illustrated in Fig. 5. The rows of this rotation matrix can be interpreted as unit vectors defining a new, transformed coordinate system. These unit vectors are marked black (Fig. 5, top). All points in the original coordinate system are rotated by the same angles in the opposite direction, relative to the new coordinate system (Fig. 5, bottom).

By applying this logic in reverse, we can derive a rotation matrix based on any three orthonormal vectors in 3D space. This matrix transforms points from the original coordinate system to a new coordinate system defined by these orthonormal vectors as unit vectors. The three vectors can be computed from any three non-collinear points in 3D space. By utilizing a consistent choice of non-collinear 3D points and orthonormal vector construction, we introduce a way to derive 3D orientation information from three 3D points. In this context, we utilize the INDEX_MCP, PINKY_MCP, and WRIST landmarks. The first vector is calculated as the vector extending from the midpoint of the triangle formed by these landmarks to the INDEX_MCP point:

$$\boldsymbol{i}'_1 = \boldsymbol{p}_{index} - \boldsymbol{p}_{midpoint} \tag{5}$$

where the midpoint can be calculated as:

$$\boldsymbol{p}_{midpoint} = \frac{1}{3}(\boldsymbol{p}_{index}+\boldsymbol{p}_{pinky}+\boldsymbol{p}_{wrist}) \tag{6}$$

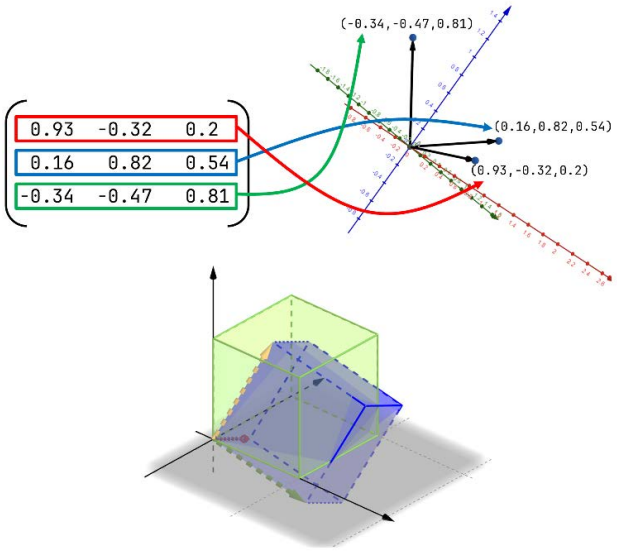

Fig. 5. Example of a rotation matrix that rotates points by 30°, 20° and 10° along the x, y and z axes, respectively.

For the second vector, we can use the normal of this triangle, which can be expressed as the cross product of any two edge vectors of the triangle:

$$\boldsymbol{j}'_1 = (\boldsymbol{p}_{index}-\boldsymbol{p}_{wrist}) \times (\boldsymbol{p}_{pinky}-\boldsymbol{p}_{wrist}) \tag{7}$$

Finally, the third vector can be calculated as the cross product of the first two vectors (see Fig. 6):

$$\boldsymbol{k}'_1 = \boldsymbol{i}'_1 \times \boldsymbol{j}'_1 \tag{8}$$

Additionally, to make these vectors unit vectors, we perform normalization by dividing them by their intensity:

$$\boldsymbol{i}_1 = \frac{\boldsymbol{i}'_1}{\|\boldsymbol{i}'_1\|}, \boldsymbol{j}_1 = \frac{\boldsymbol{j}'_1}{\|\boldsymbol{j}'_1\|}, \boldsymbol{k}_1 = \frac{\boldsymbol{k}'_1}{\|\boldsymbol{k}'_1\|} \tag{9}$$

The three vectors, calculated based on three non-collinear points, are shown in Fig. 6.

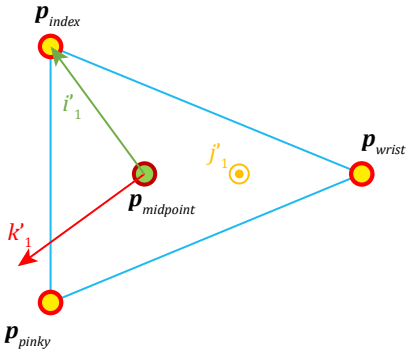

Fig. 6. The vectors i'_1, j'_1 and k'_1 calculated based on three hand landmark points: INDEX_MCP, PINKY_MCP and WRIST.

To determine the final rotation matrix to be applied to the 3D watch model, we first derive the rotation matrix that represents the orientation of the hand, using the previously described method and transpose it to get the rotation matrix that needs to be applied to the watch model. The final rotation matrix can be represented as:

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{i}_{1x} & \boldsymbol{j}_{1x} & \boldsymbol{k}_{1x} & \boldsymbol{0} \\ \boldsymbol{i}_{1y} & \boldsymbol{j}_{1y} & \boldsymbol{k}_{1y} & \boldsymbol{0} \\ \boldsymbol{i}_{1z} & \boldsymbol{j}_{1z} & \boldsymbol{k}_{1z} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{1} \end{bmatrix} \tag{10}$$

C. Smoothening Filters for 3D Landmark Stabilization

To stabilize the detected 3D keypoints over successive frames, and produce smooth 3D watch model rendering, we propose using a combination of different smoothening filters, including average, median and exponential filtering [32].

The average filter returns the average of the values inside a window. The window size is set during the filter initialization. To achieve higher performance and a time complexity of $O(1)$ in the process of smoothening, we implement this filter as a circular memory buffer. This way we avoid the array shift operation. To avoid calculating the sum of all elements in the window, we keep the previous sum. The average filter results in very smooth movement, which can be useful for the rotation, but is very sensitive to extreme values.

The exponential filter implements the exponential smoothening algorithm:

$$s_0 = x_0 \tag{11}$$

$$s_t = \alpha x_t + (1 - \alpha) s_{t-1} \tag{12}$$

where x_t is the input value, s_t the smoothened output value at time t and α the smoothening factor. The exponential filter is used to reduce small amount of degradation, which can be useful for smoothening the position.

The median buffer returns the median of all the values inside the window. The window size is set during the process of buffer initialization. Since the median calculation requires sorting the values, or maintaining a sorted array, both of which can be expensive operations in terms of time complexity, it is recommended to use a small window size to avoid performance issues in real time applications. The median filter is mostly used to filter out extreme values and can be used in combination with other filters if needed.

A comparison of different types of buffers is given in Section V, using an example of a detected 3D hand landmarks input sequence.

D. Rendering of the Watch Model

When rendering the watch model, it is essential to clip the model so that only the portion not obscured by the wrist is visible. A straightforward approach to achieve this is to utilize the far clipping plane. By positioning the watch model such that its center remains aligned with the far clipping plane at all times, the model is effectively bisected, rendering only one half visible to the user. While this clipping method may not be entirely precise, it produces realistic results. To implement this clipping technique, we employ an orthographic projection with the following parameters:

$$(x_{left}, x_{right}) = (0.5, 0.5) \tag{13}$$

$$(y_{top}, y_{bottom}) = \left(\frac{h}{2w}, -\frac{h}{2w}\right) \tag{14}$$

$$(z_{near}, z_{far}) = (-0.5, 0) \tag{15}$$

where h and w are the height and width of the input camera feed, respectively (see Fig. 7).

It is important to note that the center of the 3D watch model is constantly maintained at $z = 0$. While this approximation may not be entirely precise, it yields results that are reasonably close to the ideal position.

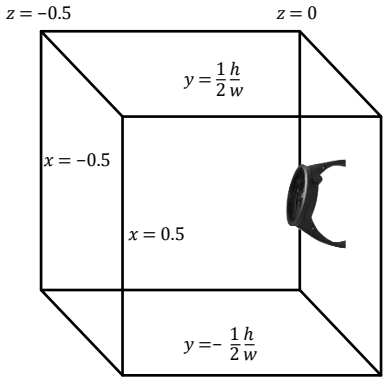

Fig. 7. The parameters of the orthographic projection: The center of the watch model remains on the far clipping plane so that only one part of the watch model is visible.

IV. Web Interface for Virtual Try-On

The proposed framework is implemented as a prototype web application using plain JavaScript, enhancing its flexibility by avoiding reliance on specific framework dependencies and facilitating seamless integration into existing websites. For hand detection, the MediaPipe tasks-vision library is employed, while the Three.js [33], [34] software library is used for manipulating the watch model in 3D space and rendering it onto an HTML canvas. The UML diagram illustrating the proposed implementation is presented in Fig. 8.

The proposed system is organized into multiple independent modules that collaboratively deliver the final virtual try-on experience. These modules include:

HandTracker Class: This class encapsulates functionalities for hand pose estimation, including hand detection, tracking, and landmark prediction. It provides methods to initialize the machine learning model (the initialize method), pause and resume hand pose estimation (the togglePredictions() method), and check the model's status (the isModelRunning() method). Additionally, a static method is available to draw the estimated landmarks onto an HTML canvas.

World Class: This class manages the manipulation and rendering of the 3D watch model. It interacts with the three core elements of a Three.js application—the scene, the camera, and the renderer. The World class provides an interface to load the watch model based on its metadata (the loadWatch() method), position, rotate, and scale the model, add lighting to the 3D scene (the setTheLights() method), and render the scene onto an HTML canvas (the render method).

ModelMetadata Class: This data class contains all necessary information for working with available watch models, including the path to the model file, the display name (title), the thumbnail, and the position, rotation, and scale offsets.

Buffer Interface: This interface defines the smoothen method. Classes implementing this interface—namely, AverageBuffer, MedianBuffer, and ExponentialBuffer—smooth the streams of position and rotation parameters, resulting in a smoother visual output of the watch model without jitter.

TryOnSystem Class: This class integrates all previously described modules into a fully functional system and is utilized in the main module of the JavaScript application.

When working with 3D models stored in files, these models may have varying positions, rotations, and scales in their local coordinate systems. To ensure a uniform interface across multiple models, we incorporate position, rotation, and scale offsets within the model metadata. When loading a watch model, we apply these initial transformations using the scene graph structure provided by the Three.js library, specifically through the World class in the loadWatch() method. This approach allows our system to maintain independence from the local coordinate systems in which the models are defined.

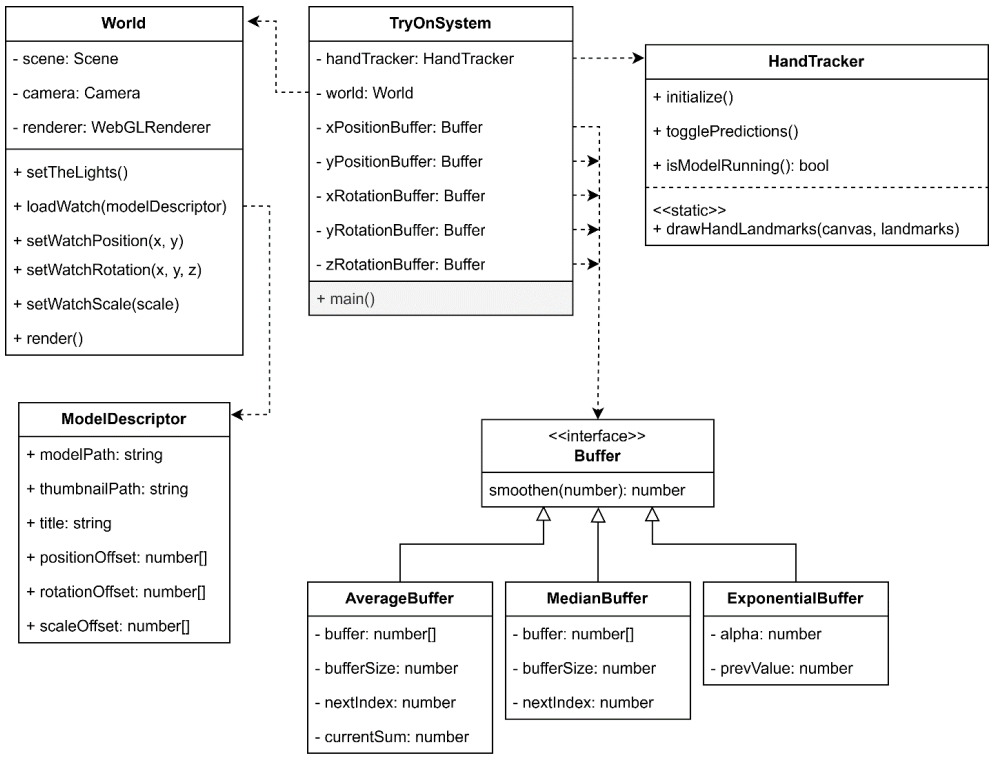

Fig. 8. The UML class diagram of the proposed virtual wristwatch try-on demo application.

V. Experimental Evaluation

A. Qualitative Evaluation

Virtual try-on systems lack a standard objective metric for image quality, making direct method comparisons difficult. In practice, we follow the approach common in generative image research: human evaluators rate the realism of the rendered images [35], [36]. Fig. 9 showcases the visual results of the virtual watch try-on generated using the proposed framework. The results demonstrate the robustness of our system in handling diverse hand poses, including complex and naturalistic gestures. Additionally, the system consistently achieves a realistic virtual try-on effect, even under practical scenarios involving varying lighting conditions, occlusions, and hand orientations.

Crucially, however, there is no public benchmark for AR wristwatch try-on quality. Existing datasets (e.g., VITON [6][37]) focus on clothing, while recent research on accessories such as watches and jewelry [14] explicitly note that they "were unable to find a publicly available dataset". Prior studies have therefore resorted to small-scale subjective evaluations (often informal volunteer tests) and typically have not published any evaluation dataset [11], [12], [13].

To address these gaps, we created a new dataset and conducted a controlled user study. We collected 120 wrist images, where, for each image we store the frame without the AR watch, the frame with the AR rendered watch and a mask indicating the watch position. We then ran a perceptual study with N=10 volunteers from our university. In each trial, a participant was shown an image of a person wearing the augmented wristwatch and was asked to rate the visual quality of the try-on result on a scale from 1 (poor) to 5 (excellent). The resulting mean opinion score (MOS) was 4.35 out of 5, indicating that our augmented images were generally perceived as quite realistic. The full dataset can be provided for research purposes on demand.

B. Performance Testing

The performance evaluation of the 3D virtual wristwatch try-on framework demonstrates its capability for fast and efficient real-time operation. The developed prototype was tested on several platforms, including edge devices such as mobile phones (iPhone 15 Pro and Samsung S24 Ultra) and a mainstream PC equipped with an Intel Core i5 CPU with integrated graphics. Performance was measured over 1,000 consecutive frames to assess the system's ability to maintain real-time responsiveness. For each metric we performed N=5 independent measurements under the same conditions, calculated the mean value, and then computed the standard deviation.

TABLE II. Time Performance Evaluations With Standard Deviation on Several Platforms

	iPhone 15 Pro	Samsung S24 Ultra	PC
Hand Landmark Detection [ms]	10.11 ± 2.43	24.46 ± 3.26	15.66 ± 1.9
Rendering Time [ms]	1.45 ± 0.58	2.9 ± 1.18	2.56 ± 0.37
Total Processing Time [ms]	11.56 ± 2.51	27.36 ± 3.43	18.22 ± 1.93

The results, presented in Table II, demonstrate that the system is highly efficient and suitable for in-browser execution, with all computationally intensive tasks, including hand pose estimation and 3D rendering, executed locally on the device. On all tested platforms, the combined inference and rendering times consistently remain within the threshold required for real-time performance in AR systems [37], with the measured total standard deviation of up to 3.43 milliseconds also falling within the acceptable range, as this variation is imperceptible to the human eye. These results underscore the system's efficiency and suitability for edge-based interactive applications making the system particularly suitable for web-based augmented reality applications and highlighting its potential to deliver interactive real-time user experience.

The system employs three types of buffers: average, median, and exponential, to achieve smooth transitions in the virtual wristwatch positioning. The average buffer computes the mean of recent values,

Fig. 9. Preview of the visual results of the proposed framework in a real-world environment for various hand positions and orientations, using different 3D watch models.

providing a steady smoothing effect, but it can be slow to adapt to sudden changes, leading to slight lag. The median buffer, on the other hand, is robust to outlier values, but less effective in capturing fine-grained variations. The exponential buffer applies a weighted smoothing where recent values are given more importance, allowing for quicker responsiveness to changes while still maintaining a degree of stability, though it may amplify short-term noise. A comparison of the capabilities of these buffers is shown on Fig. 10.

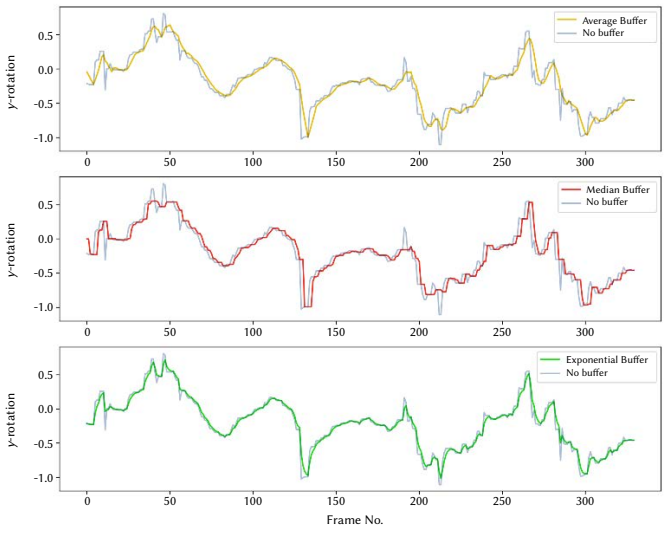

Fig. 10. Comparison of different buffers used as smoothing filters: average, median, and exponential buffers. The change in the watch's y-rotation value (in radians) is shown on the y-axis, across 330 consecutive frames of hand movement on the x-axis.

C. Limitations

While the 3D virtual wristwatch try-on system offers an innovative approach to augmenting reality through real-time on-device hand pose estimation, it is not without limitations. The accuracy of the virtual wristwatch positioning is directly dependent on the estimated hand pose. When the hand is oriented or bent in an unconventional way relative to the forearm, the calculated watch transform may appear unnatural or misaligned. This occurs because the system assumes a consistent and anatomically expected relationship between the hand and the forearm. However, this assumption simplifies computation and ensures stable placement under typical poses. Extreme poses or unusual orientations, as illustrated in Fig. 11, highlight this limitation, where the watch appears to "float" or distort in relation to the wrist. Furthermore, if the user's hand is significantly occluded, out of the camera's field of view, or obscured by external objects, the hand pose cannot be precisely detected, resulting in a loss of functionality under such circumstances. In practice, this means that the model responds well to standard, unobstructed hand orientations but may fail to produce stable and realistic overlays in edge cases. Additionally, the current model does not fully account for individual differences in wrist shape and size, which may impact the perfect alignment of the virtual watch in some cases. These limitations underscore the need for additional contextual inputs, such as arm tracking, to improve robustness under challenging conditions. Furthermore, the empirical evaluation is based on a dataset that may not fully capture the diversity of hand poses, skin tones, and environmental conditions encountered in real-world usage. At present, however, no public benchmark exists for assessing AR accessory try-on quality, as described in Section V.A.

Fig. 11. Visualization of extreme hand poses and unusual orientations, relative to the forearm, demonstrating the limitation where the watch appears to "float" or distort in relation to the wrist.

VI. Conclusion

In this paper, we proposed the design and implementation of an edge-centric augmented reality framework for virtual wristwatch try-ons as a web application. The core functionality of our system includes calculating the watch positioning based on only three hand landmarks powered by on-device machine learning, eliminating the need for markers or expensive cloud-based processing. By leveraging a web-based format, our framework can be seamlessly integrated into existing e-commerce platforms, making it widely accessible to users without requiring specialized hardware or software.

The paper presented a lightweight, edge-executable pipeline for watch try-on. It achieved interactive frame rates (>30 fps) and high visual quality, with a mean opinion score of 4.35. The results demonstrated the feasibility of highly realistic, immersive AR try-on for small accessories on edge devices and provided a viable alternative to cloud-based solutions. While limitations exist, such as inaccuracies in watch placement during uncommon hand orientations or significant occlusions, these scenarios are rare in typical watch try-on scenarios. Given its robust real-time performance and ease of integration, our framework has significant potential to advance marketing and sales strategies in the watch industry, offering a scalable and engaging tool to enhance the online shopping experience for consumers.

CRediT Authorship Contribution Statement

Stevica Cvetković: Writing – review & editing, Writing – original draft, Resources, Methodology, Investigation, Conceptualization, Supervision.

Matija Špeletić: Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Investigation.

Jelena Nikolić: Writing – review & editing.

Data Statement

The data supporting the findings of this study are available from the corresponding author upon request.

Declaration of Conflicts of Interest

No conflict of interest exists.

Acknowledgment

This work has been supported in part by the Horizon Europe Twinning project AIDA4Edge (Grant Agreement No. 101160293), and by the Ministry of Science, Technological Development and Innovation of the Republic of Serbia under Grant No. 451-03-34/2026-03/200102.

References

[1] J. Kim and S. Forsythe, "Adoption of Virtual Try-on technology for online apparel shopping," *Journal of Interactive Marketing*, vol. 22, no. 2, pp. 45–59, 2008, doi: https://doi.org/10.1002/dir.20113.

[2] H. Hwangbo, E. Kim, S.-H. Lee, and Y. J. Jang, "Effects of 3D Virtual 'Try-On' on Online Sales and Customers' Purchasing Experiences," *IEEE Access*, vol. 8, pp. 189479–189489, 2020, doi: https://doi.org/10.1109/ACCESS.2020.3023040.

[3] F. Bonetti, G. Warnaby, and L. Quinn, "Augmented Reality and Virtual Reality in Physical and Online Retailing: A Review, Synthesis and Research Agenda," in *Augmented Reality and Virtual Reality: Empowering Human, Place and Business*, T. Jung and M. C. tom Dieck, Eds., Cham: Springer International Publishing, 2018, pp. 119–132. doi: https://doi.org/10.1007/978-3-319-64027-3_9.

[4] J. Scholz and K. Duffy, "We ARe at home: How augmented reality reshapes mobile marketing and consumer-brand relationships," *Journal of Retailing and Consumer Services*, vol. 44, pp. 11–23, 2018, doi: https://doi.org/10.1016/j.jretconser.2018.05.004.

[5] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge Computing: Vision and Challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016, doi: https://doi.org/10.1109/JIOT.2016.2579198.

[6] X. Han, Z. Wu, Z. Wu, R. Yu, and L. S. Davis, "VITON: An Image-Based Virtual Try-on Network," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7543–7552. doi: https://doi.org/10.1109/CVPR.2018.00787.

[7] B. Wang, H. Zheng, X. Liang, Y. Chen, L. Lin, and M. Yang, "Toward Characteristic-Preserving Image-based Virtual Try-On Network," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Sep. 2018.

[8] M. R. Minar, T. T. Tuan, H. Ahn, P. Rosin, and Y.-K. Lai, "CP-VTON+: Clothing Shape and Texture Preserving Image-Based Virtual Try-On," in *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, Jun. 2020.

[9] Z. Chong and L. Mo, "ST-VTON: Self-supervised vision transformer for image-based virtual try-on," *Image and Vision Computing*, vol. 127, p. 104568, 2022, doi: https://doi.org/10.1016/j.imavis.2022.104568.

[10] L. Duan, Z. Yueqi, W. Ge, and P. Hu, "Automatic three-dimensional-scanned garment fitting based on virtual tailoring and geometric sewing," *Journal of Engineered Fibers and Fabrics*, vol. 14, p. 155892501882531, Feb. 2019, doi: https://doi.org/10.1177/1558925018825319.

[11] F. Wu and A. Dellinger, "Real-Size Experience for Virtual Try-On," *Computer Science Research Notes*, vol. 3401, no. 1, pp. 307–314, 2024, doi: https://doi.org/10.24132/CSRN.3401.33.

[12] M. M. Mohan, "Ar Watch Try - on Application for Android Devices," *Journal of Science Technology and Research (JSTAR)*, vol. 2, no. 1, pp. 39–48, 2021.

[13] S. Gupta, M. Pahwa, P. Gupta, and S. Kaur, "ARZARA: Augmented reality app to try watch on your wrist," *Fusion: Practice and Applications*, pp. 50–56, 2020, doi: https://doi.org/10.54216/FPA.020202.

[14] T.-Y. Chang and S. K. Lekena, "GlamTry: Advancing Virtual Try-On for High-End Accessories." 2024. [Online]. Available: https://arxiv.org/abs/2409.14553

[15] H. Tang, B. Ren, P. Wu, and N. Sebe, "Hierarchical Cross-Attention Network for Virtual Try-On," *IEEE Transactions on Multimedia*, vol. 27, pp. 4454–4466, 2025, doi: https://doi.org/10.1109/TMM.2025.3548437.

[16] H. Song, E. Baek, and H. Choo, "Try-on experience with augmented reality comforts your decision: Focusing on the roles of immersion and psychological ownership," *Information Technology and People*, vol. 33, no. 4, pp. 1214–1234, 2020, doi: https://doi.org/10.1108/ITP-02-2019-0092.

[17] J.-H. Kim, M. Kim, M. Park, and J. Yoo, "Immersive interactive technologies and virtual shopping experiences: Differences in consumer perceptions

between augmented reality (AR) and virtual reality (VR)," *Telematics and Informatics*, vol. 77, p. 101936, 2023, doi: https://doi.org/10.1016/j.tele.2022.101936.

[18] V. Arya, R. Sambyal, A. Sharma, and Y. K. Dwivedi, "Brands are calling your AVATAR in Metaverse–A study to explore XR-based gamification marketing activities & consumer-based brand equity in virtual world," *Journal of Consumer Behaviour*, vol. 23, no. 2, pp. 556–585, 2024, doi: https://doi.org/10.1002/cb.2214.

[19] J. Galán, C. García-García, F. Felip, and M. Contero, "Does a presentation Media Influence the Evaluation of Consumer Products? A Comparative Study to Evaluate Virtual Reality, Virtual Reality with Passive Haptics and a Real Setting," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 6, no. 6, pp. 196–207, 2021, doi: https://doi.org/10.9781/ijimai.2021.01.001.

[20] C. Osimani, J. J. Ojeda-Castelo, and J. A. Piedra-Fernandez, "Point Cloud Deep Learning Solution for Hand Gesture Recognition," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, no. 4, pp. 78–87, 2023, doi: https://doi.org/10.9781/ijimai.2023.01.001.

[21] L. Hui, L. Huayang, Z. Wei, and L. Hao, "The Human Motion Behavior Recognition by Deep Learning Approach and the Internet of Things," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 8, no. 7, pp. 55–65, 2024, doi: https://doi.org/10.9781/ijimai.2024.07.004.

[22] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "Mediapipe hands: On-device real-time hand tracking," *ArXiv Prepr. ArXiv200610214*, 2020.

[23] A. Mujahid, M. J. Awan, A. Yasin, M. A. Mohammed, R. Damaševičius, R. Maskeliūnas, and K. H. Abdulkareem, "Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model," *Applied Sciences*, vol. 11, no. 9, 2021, doi: https://doi.org/10.3390/app11094164.

[24] S. Han, B. Liu, R. Cabezas, C. D. Twigg, P. Zhang, J. Petkau, T.-H. Yu, C.-J. Tai, M. Akbay, Z. Wang, *et al.*, "MEgATrack: monochrome egocentric articulated hand-tracking for virtual reality," *ACM Transactions on Graphics*, vol. 39, no. 4, 2020, doi: https://doi.org/10.1145/3386569.3392452.

[25] S. Cvetkovic, N. Savic, and I. Ciric, "Deep Transfer Learning Approach for Robust Hand Detection," *Intelligent Automation and Soft Computing*, vol. 36, no. 1, pp. 967–979, 2023, doi: 10.32604/iasc.2023.032526.

[26] I. Rehman, S. Ullah, and M. Raees, "Two Hand Gesture Based 3D Navigation in Virtual Environments," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 5, no. 4, pp. 128–140, 2019, doi: https://doi.org/10.9781/ijimai.2018.07.001.

[27] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, F. Zhang, C.-L. Chang, M. Yong, J. Lee, and others, "Mediapipe: A framework for perceiving and processing reality," in *Third workshop on computer vision for AR/VR at IEEE computer vision and pattern recognition (CVPR)*, 2019.

[28] *Three.js*. [Online]. Available: https://github.com/mrdoob/three.js/

[29] J. Nikolić, Z. Perić, D. Aleksić, S. Tomić, and A. Jovanović, "Whether the Support Region of Three-Bit Uniform Quantizer Has a Strong Impact on Post-Training Quantization for MNIST Dataset?," *Entropy*, vol. 23, no. 12, 2021, doi: https://doi.org/10.3390/e23121699.

[30] D. M. Jiménez-Bravo, Á. L. Murciego, A. S. Mendes, L. A. Silva, and D. H. D. L. Iglesia, "Edge Face Recognition System Based on One-Shot Augmented Learning," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol. 7, no. 6, pp. 31–44, 2022, doi: https://doi.org/10.9781/ijimai.2022.09.001.

[31] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Second. Cambridge University Press, ISBN: 0521540518, 2004.

[32] J. S. Simonoff, *Smoothing Methods in Statistics*, 1st ed. in Springer Series in Statistics. Springer New York, NY.

[33] A. Anyuru, *Professional WebGL programming: developing 3D graphics for the Web*. John Wiley & Sons, 2012.

[34] G. Lavoué, L. Chevalier, and F. Dupont, "Streaming compressed 3D data on the web using JavaScript and WebGL," in *Proceedings of the 18th international conference on 3D web technology*, 2013, pp. 19–27.

[35] Q. Chen and V. Koltun, "Photographic image synthesis with cascaded refinement networks," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 1511–1520.

[36] C. Gan, Z. Gan, X. He, J. Gao, and L. Deng, "StyleNet: Generating Attractive Visual Captions With Styles," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017.

[37] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Real-Time Detection and Tracking for Augmented Reality on Mobile Phones," *IEEE Transactions on Visualization and Computer Graphics*, vol. 16, no. 3, pp. 355–368, 2010, doi: https://doi.org/10.1109/TVCG.2009.99.

Stevica Cvetković

Dr. Stevica Cvetković is an Associate Professor at the Faculty of Electronic Engineering, University of Niš, Serbia. He completed his BSc and MSc studies in Computer Science at the University of Niš and subsequently engaged in research collaborations with the Polytechnic University of Madrid and the Technical University of Munich. He earned his Ph.D. in 2018 in the field of Computer Vision. Dr. Cvetković is the founder and head of the Laboratory for Visual Technologies at the Faculty of Electronic Engineering, University of Niš. His research interests include Deep Learning, Computer Vision, and Generative Artificial Intelligence. He has authored over 60 scientific publications, including 10 articles in leading international journals.

Matija Špeletić

Matija Špeletić is a Ph.D. student and a Teaching Assistant at the Faculty of Electronic Engineering, University of Niš, Serbia. He completed his BSc studies at the University of Niš with honors and earned his MSc degree in Artificial Intelligence in 2024. His research has been focused on Deep Learning and Computer Vision, which were also the topics of his bachelor and master theses. His current research interests include Generative Artificial Intelligence, Deep Learning, Computer Vision.

Jelena Nikolić

Jelena Nikolić is a Full Professor at the Faculty of Electronic Engineering, University of Niš. She received her B.Sc., M.Sc., and Ph.D. degrees in Telecommunications in 2003, 2006, and 2011, respectively, from the Faculty of Electronic Engineering, University of Niš. She is the author of 135 papers, including 59 in journals with IF, and for her outstanding scientific publications, she was awarded by Telenor foundation in 2017. She has served as a reviewer for numerous journals and textbooks. She is a lecturer on several subjects from the Data Compression and AI field. Her research interests include Artificial Intelligence, Compression of Neural Networks, Edge Computing.