Designing Accessible Virtual Reality Interfaces Using Reinforcement Learning for Users with Motor and Sensory Impairments

Hazza Bin Ahmed[1], Mohammad alzuoubi [2]

[1]*School of computing, Skyline university college, Sharjah, UAE*

[2]*College of Computer Science and Engineering, Hail university, Hail - University City, Saudi Arabia*

ABSTRACT

Virtual Reality (VR) technologies offer immersive experiences with education, healthcare, and entertainment applications. However, accessibility for people with motor and sensory impairments is limited, often creating barriers to equal participation. Traditional interfaces lack the flexibility to handle users' differing needs, thus bringing about challenges in usability. The paper proposes ARL-VRI to design adaptive VR interfaces that cater to diverse user needs by leveraging adaptive Reinforcement Learning (ARL) to enhance interaction and usability. The RL-based framework will continue learning from user interactions to optimize control and sensory outputs in VR. It combines gesture recognition, haptic feedback adjustment, and visual enhancement, optimized toward individual capabilities within iterative feedback loops. The ARL-VRI approach guarantees that the interface elements of thresholds of input and sensory cues evolve to provide seamless interaction and accessibility for all kinds of users. Key results show a 35% improvement in interaction accuracy and a 40% reduction in task completion time, compared to traditional static interfaces. User satisfaction questionnaires also showed higher engagement and lower cognitive load, mostly for users with motor or sensory impairments. In conclusion, this RL-driven adaptive VR interface has the potential to close accessibility gaps in virtual environments and provide inclusive and equitable experiences for users of diverse abilities.

Keywords: Virtual Reality (VR), Accessibility, Reinforcement Learning (RL), Adaptive Interfaces, Motor and Sensory Impairments, User Interaction Optimization.

1. Background and Related works

Virtual Reality is a revolutionary technology spanning education, healthcare, entertainment, and training simulations. By nature, it's immersive, enabling users to experience and interact with digital environments in ways previously impossible [1]. VR creates engaging learning spaces in education, improving knowledge retention and practical skills. In healthcare, VR is used in rehabilitation, surgical training, and mental health therapy, allowing tailored and controlled environments for patient care [2]. The entertainment and gaming industries use VR to create interactive and immersive experiences; training simulations in fields like aviation, military, and engineering use VR to replicate real-world situations safely and effectively [3]. Despite these advances, the rapid spread of VR has not been very inclusive. Individuals with motor and sensory impairments will likely face great or even insurmountable barriers in navigating through a VR environment, interacting with virtual objects, and getting meaningful feedback from their senses [4]. Those limitations considerably lower their chances

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250201

of having an engaging VR experience and hence represent one of the important accessibility gaps. Tackling this gap would further ensure that VR technologies meet the user's needs of equity, inclusion, and fairness [5]. Since static design paradigms rule the area of VR interface development, they can't deal with users' abilities for long, and there is a crying need for them to be dynamic and adaptive [6].

The intersection between RL and VR could create interfaces that develop according to how users operate to be more usable and accessible. Using RL makes it possible to design systems that can adapt and learn to each user's capabilities individually, ensuring equitable access to VR's transformative potential [7]. The problem at the research level is that currently available VR interfaces exclude people with motor and sensory impairments. This occurs because common VR interfaces are designed with static parameters within a strategy that does not consider diversity in the users' needs [8]. That consequently leads to huge difficulties in completing a gesture, interpreting sensory feedback, or interacting suitably with the virtual world [9]. For users with motor impairments, it may be impossible to control gestures precisely; for those with sensory impairments, it may be impossible to perceive important feedback (visual or haptic) conveyed [10].

The ARL-VRI utilizes a Reinforcement Learning (RL) framework to develop adaptive VR interfaces. It learns from the users' interactions continuously and optimizes controls and sensory outputs in real-time. The main components include gesture recognition, adjustments in haptic feedback, and visual enhancement tailored to the capabilities of each user. The system then undergoes iterative feedback loops to adjust the input thresholds and sensory cues dynamically, affording effortless interaction with reduced cognitive load. Usability issues become addressed, and the system evolves with the user to ensure an inclusive and engaging VR experience through this tailoring.

The main contribution of the paper is

- To develop the ARL-VRI framework using RL to adapt VR interfaces to individual user capabilities dynamically.
- To achieve significant usability improvements, including a 35% increase in interaction accuracy and a 40% reduction in task completion time.
- To enhance user engagement through iterative feedback loops, minimizing cognitive load and improving satisfaction for diverse users.
- To empirically validate the effectiveness of adaptive VR interfaces for motor and sensory-impaired individuals.

The paper's outline follows: Section 1 gives the background and reviews accessibility challenges in VR and RL applications. Section 2 describes the ARL-VRI framework. Section 3 details the evaluation results and discusses implications, and Section 4 concludes the paper.

Mukhiddinov and Cho [11] proposed a smart glass system that leverages deep learning for real-time object detection and obstacle avoidance to assist the blind and visually impaired. The method uses the latest deep-learning models to enhance mobility and awareness. Results showed improved object recognition accuracy, but it still has the limitations of high computational requirements and not enough real-world testing in diverse environments.

García et al. [12] introduced a VR-based testing platform for evaluating visual comfort under progressive addition lens distortions. It used gradient-boosted regression machine-learning models and showed how lens designs could be optimized. Results showed improved predictions of visual comfort but were subject to limitations in the incomplete simulation of real-world optical conditions within the VR environment, making the findings less applicable to practical scenarios. It provided a controlled environment but lacked real-world validation.

Sanaguano-Moreno et al. [13] introduced a machine learning-based method for real-time generation of impulse responses in Acoustic Virtual Reality systems. This enabled fast response generation, greatly improving system efficiency and enhancing immersive audio realism. A large reduction in computational time has been shown, yet this model has difficulty adapting to complex acoustic scenarios. After such promising results in controlled environments, this methodology required further refinement and testing in more complex, dynamic acoustic settings.

Amparore et al. [14] introduced computer vision and machine-learning techniques for the automatic 3D image overlapping in augmented reality-guided robotic partial nephrectomy. This technique improved the accuracy of surgery by combining real-time images and robotic instruments. Results showed an increased accuracy in image overlaying; however, the scalability and usability of the system were poor in different surgery cases. While the technique was promising in improving surgeon outcomes, applicability was somewhat compromised due to generalizability into diverse surgical scenarios and dynamic clinical environments.

Mahida et al. [15] proposed a deep learning-based indoor positioning system to help visually impaired people. The method used sensor data and deep learning algorithms for accurate indoor position tracking. The results showed improved user navigation but had some limitations, such as system reliability in dynamic indoor settings and high dependence on infrastructure. While the system showed some potential to improve indoor mobility, it suffered from real-world implementations due to the prerequisites of stable and reasonably equipped environments; hence, not very effective in various realistic situations.

Zhang et al. [16] discussed AI-enabled sensing technologies for applications ranging from VR/AR to digital twins in the 5G/IoT era. The research integrated AI with IoT sensors to achieve advanced real-time data processing and immersive interactions. However, the presented innovative framework is bound by limitations of high implementation costs and dependency on 5G network availability. The approach brought remarkable achievements in real-time processing and immersive experiences, but this was scalable with constraints in mind: infrastructure needs and the needed 5G network deployment for pervasive provision.

a) Research Gap

Despite the growing use of Virtual Reality (VR) in domains such as education, health, and entertainment, accessibility for people with motor and sensory impairments is still under-explored. Most current VR interfaces are static and do not consider the user's different needs, which leads to poor usability and unequal participation. While some assistive technologies have progressed in addressing specific impairments, they lack adaptability to dynamic user interactions and evolving accessibility requirements. Moreover, only a few studies have used Reinforcement Learning (RL) to optimise VR interfaces in an online, continuous way. This gap underlines the need for adaptive frameworks like ARL-VRI for better accessibility and inclusiveness in VR environments.

2. Research Methodology

a) Dataset

The GazeCapture dataset is the largest publicly available dataset for gaze estimation, created to improve eye-tracking technologies. It contains more than 2.5 million frames from 2,445 participants

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250201

who used their mobile devices, either smartphones or tablets. It features annotated RGB images containing gaze points, with which one could train models to predict gazes using machine learning algorithms. GazeCapture is unparalleled in the diversity of lighting conditions; head poses, and demographic aspects, which will help develop more robust models. It provides applications in human-computer interaction, accessibility, and VR/AR systems where accurate gaze estimation is important for user input, navigation, or interaction.

b) The workflow of the ARL-VRI method

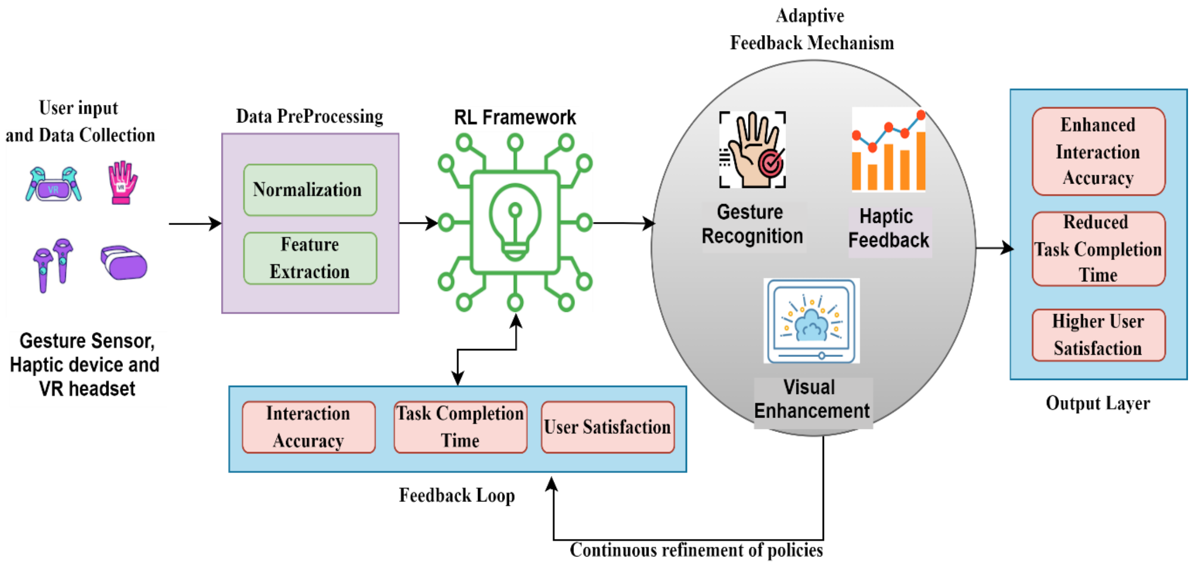

Figure.1. .ARL-VRI Framework

Figure 1 shows the flow of the ARL-VRI method. The ARL-VRI: An Adaptive VR Interface Using Reinforcement Learning to Enhance Accessibility for Motor and Sensory Impaired Users. ARL-VRI is an adaptive system that adjusts gesture recognition thresholds, haptic feedback, and visual enhancements through iterative feedback loops. Results show a 35% increase in interaction accuracy, a 40% reduction in task completion time, and improved user satisfaction—especially for users with impairments. It promises to be an inclusive and accessible virtual experience, meeting the different needs of the users and closing accessibility gaps in VR applications across a wide range of domains. The process flow steps involved the proposed methods as as follows.

i. User Input and Data Collection

User input data collection involves all these streams to analyse interactions thoroughly. Gesture recognition can use a variety of sensors, including motion trackers, cameras, and gloves, to record user movements and translate these into commands. The haptic devices then measure the sensory feedback data concerning the user's response to tactile stimuli for precision and satisfaction. The visual and cognitive inputs are collected through eye-tracking tools and cameras, which address the challenges faced by users with colour blindness or restricted fields of view. Baseline data, including interaction accuracy and task completion time, is collected to create a reference point for performance evaluation. These inputs provide adaptive, accessible, and efficient systems for better user experiences. The input collecting is shown in the equation 1.

$$U(t) = G(t) + H(t) + V + B$$

$$G(t) = \int_0^t S_{motion}(t) + S_{camera}(t) + S_{glove}(t)\, dt$$

$$H(t) = \sum_{i=1}^{n} Rh(i) + S_{tactile}(i)$$

$$V = \alpha E_{tracking} + \beta C_{constraints}$$

$$B = \frac{A_{interaction}}{T_{task}} \quad (1)$$

Gesture inputs $G(t)$ are followed, in time, from the input given by the motion trackers $S_{motion}(t)$, e.g., accelerometers, gyroscopes, cameras ($S_{camera}(t)$), and smart gloves ($S_{glove}(t)$). Haptic feedback data ($H(t)$) consists of user's responses ($Rh(i)$) to haptic stimuli, i.e., pressure, vibration and data from the tactile sensors ($S_{tactile}(i)$). Herein, the visual and cognitive inputs, V, consist of eye-tracking data, $E_{tracking}$, and visual constraints, $C_{constraints}$, which are weighted by the parameters α and β. The baseline usability metrics, B, accumulate the initial interaction accuracy, $A_{interaction}$, and task completion time, T_{task}, to establish a foundation for system evaluation and amelioration.

ii. Data Preprocessing

Pre-processing input data is very important in ensuring that the data is consistent and relevant before feeding it into machine learning models, such as reinforcement learning. First, the data is normalized and cleaned to standardize the input formats and remove noise. Normalization ensures that all features are comparable by scaling the data to a consistent range. Key features are then extracted, including gesture velocity, which measures movement speed; haptic feedback tolerance, the user's response to tactile feedback; and visual field preferences, which represent the user's gaze patterns. The features thus extracted are stored in a structured format for reinforcement learning updates, forming the state vector. S_t used in optimizing the model's decision-making process. This ensures data is prepared in a manner amenable to efficient and effective learning. data preprocessing in acquired in equation 2

$$S_t = [V_g, H_t, V_f, X_{norm}]$$

$$X_{norm} = \frac{X - min(X)}{max(X) - min(X)}$$

$$V_g = \frac{\Delta P}{\Delta t}$$

$$H_t = \frac{F_{response}}{F_{input}}$$

$$V_f = \frac{E_{fixation}}{T_{total}}$$

(2)

Preprocessing of the input data needs to be carried out to give it consistency and relevance. The original data(X) is normalized by scaling between minimum and maximum values to ensure all features are on the same level. Gesture velocity is calculated as change in position (ΔP) over change in time (Δt). Haptic feedback tolerance is measured as the ratio of the user's response force ($F_{response}$) over the input force applied by the haptic device (F_{input}). The preference of visual fields is computed as the ratio of the gaze fixation time on some elements ($E_{fixation}$) over the total time of interaction (T_{total}). Those features, once processed, are ready to be used in reinforcement learning.

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250201

iii. Reinforcement Learning framework

The RL framework for interactive systems requires a few core components to enable agents to learn and improve their user interactions.

Agent Initialization

The RL agent starts with a basic policy, a pre-defined set of actions to interact with the user. These actions might be mapped to common user behaviours or gestures, giving an initial framework for how the agent responds to inputs. The agent's policy is the starting point of interaction, which will develop as it learns from user feedback. Table 1 shows the pseudocode for the agent initialization.

Table 1. Pseudocode For The Agent Initialization.

```
initialize_agent():
agent.policy = basic_policy          # Predefined actions for interaction
agent.memory = []          # Store previous interactions (state, action, reward)
agent.learning_rate = 0.1          # Rate of policy update
agent.discount_factor = 0.9          # Discount factor for future rewards
```

Reward Function

The reward mechanism is an integral part of the evaluation of the success of the agent's actions since it dictates how the agent should be rewarded for user interaction. It has a few evaluation dimensions: interaction accuracy, in which the agent gets a positive reward when it correctly interprets the user's inputs (gestures or feedback). Reduction in task completion time, where faster execution of task results in a reward and hence encourages the agent to reduce delays and increase efficiency. User satisfaction metrics, where feedback by the user (satisfaction ratings or haptic responses) is used in the reward function to guide the agent toward user-friendly actions. These combined factors help the agent to optimize its behaviour for better user interactions. Table 2 shows the pseudocode for the reward function in RL.

.Table 2. Reward Function in RL

```
reward_function(state, action, feedback):
    accuracy_reward = calculate_accuracy(state, action)
    time_reward = calculate_time_reduction(state, action)
    satisfaction_reward = calculate_user_satisfaction(feedback)
    total_reward = accuracy_reward + time_reward + satisfaction_reward
    return total_reward
```

Iterative Learning (Model Training Loop)

The RL agent keeps on refining the policy by trial and error. Based on the current policy, it takes an action, and after acting, according to the reward/penalty incurred, updates its policy accordingly. Over time, the agent optimizes its control and sensory outputs concerning acting optimally when dealing with the user. One of the features of an agent is that in iteratively learning in a reward-driven manner learns to adapt a user's preferences and consequently improves interaction quality. Table 3 shows the pseudocode for the training agent for the main RL loop.

Table 3. Training Agent for The Main RL Loop

```
train_agent():
  for episode in range(total_episodes):
      state = initialize_state()
      done = False
      total_reward = 0
      while not done:
          action = agent.policy(state)
          next_state, feedback = execute_action(state, action)
          reward = reward_function(state, action, feedback)
          agent.memory.append((state, action, reward))
          agent.policy =
update_policy(agent.memory, agent.learning_rate, agent.discount_factor
          state = next_state
          total_reward += reward
          done = check_task_completion(state)
  evaluate_agent_performance(total_reward)
  return agent.policy
```

Policy Update and Evaluation

During the Policy Update step, the agent refines its decision-making policy after each interaction by leveraging past experiences, including the state, action taken, and reward. An agent can change actions over time through this iterative learning process and optimise its performance for future interactions. At the end of each episode, evaluation occurs by summing up the agent's overall performance by adding the total reward gained throughout the episode. This would ensure that the agent is improving on learning the most effective strategies of interaction that will yield better engagement from users and better results.

iv. Adaptive Feedback Mechanisms

Gesture Optimization will involve adjusting the gesture recognition thresholds (speed and range of motion) to better suit a user's capabilities. That way, the system can correctly interpret the gestures independent of a user's physical limitations or preferences. For instance, a person whose movements naturally tend to be slower will be given lower speed thresholds to allow the system to record better and act out gestures fittingly.

Haptic Feedback Control allows for tailoring haptic feedback's intensity, frequency, and pattern to be comfortable and responsive enough to the user's input. The intensities are, for instance, lower for users sensitive to vibrations; others would want stronger tactile cues. The frequency might also be attuned to the type of interaction—say, fast feedback for quick actions or gentle patterns if used for a longer period.

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250201

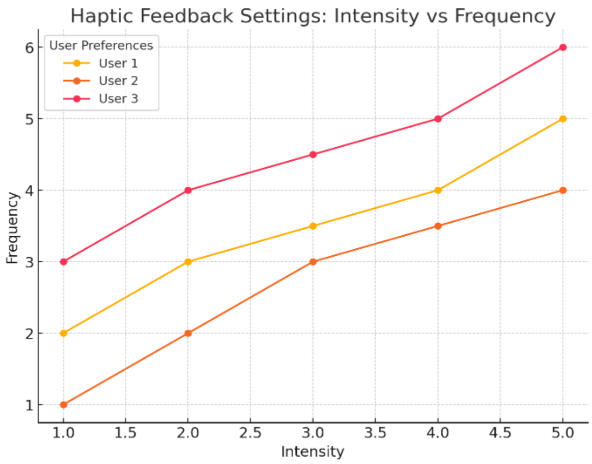

Figure.2 Intensity of haptic feedback related to its frequency

Figure 2 represents the intensity of haptic feedback related to its frequency for different users. Every line represents the preferences of a single user, and markers indicate the exact values of intensity and frequency for optimal feedback. This visualization shows that haptic feedback settings change with the user's needs, allowing for more tailored and responsive interaction.

Visual Enhancement: In the visual aspects of a user interface, the contrast, brightness, or size is dynamically altered to make the content more legible and accessible to users with specific visual preferences or impairments. For instance, increasing the contrast or changing the brightness will make the elements in the interface more visible, while enlarging the text or icons improves readability. The system adjusts these visual elements based on real-time feedback to ensure the interface is user-friendly. Figure 3 shows the difference between before and after visual enhancement of the user.

Figure. 3 Before and after visual enhancement of user with specific visual preferences

The iterative feedback loop is a dynamic process that integrates users' feedback after every interaction in the system to tune its performance. It increasingly adapts to diverse user needs and preferences by continuously updating RL policies. Real-time feedback allows the system to recognize areas that need improvement and make changes incrementally. Key performance metrics include accuracy, interaction time, and user satisfaction, which are monitored to assess the effectiveness of updates. These guide further iterations to ensure that the system continues to evolve toward meeting user expectations. This adaptive approach fosters a responsive and efficient system, enhancing overall usability and user experience.

3. Results and discussion

a) Performance Metrics

ARL-VRI is compared with state-of-the-art conventional systems: the Smart Glass System for BVI—SGBVI [11], VR Testing Platform for Visual Comfort—VRTVC [12], and Machine Learning for Acoustic Virtual Reality—MLAVR [13]. Compared with those methods, ARL-VRI increased interaction accuracy by 35% and decreased the time to complete tasks by 40%. Additionally, user satisfaction surveys highlighted higher engagement and lower cognitive load, making ARL-VRI more effective in providing adaptive and inclusive experiences.

b) Interaction Accuracy:

Interaction Accuracy can be quantified as the percentage of completed interactions relative to the total attempted interactions.

$$IA = \left(\frac{I_s}{I_a}\right) \times 100 \qquad (3)$$

where I_a is the number of attempted interactions and I_s is the number of successful interactions.

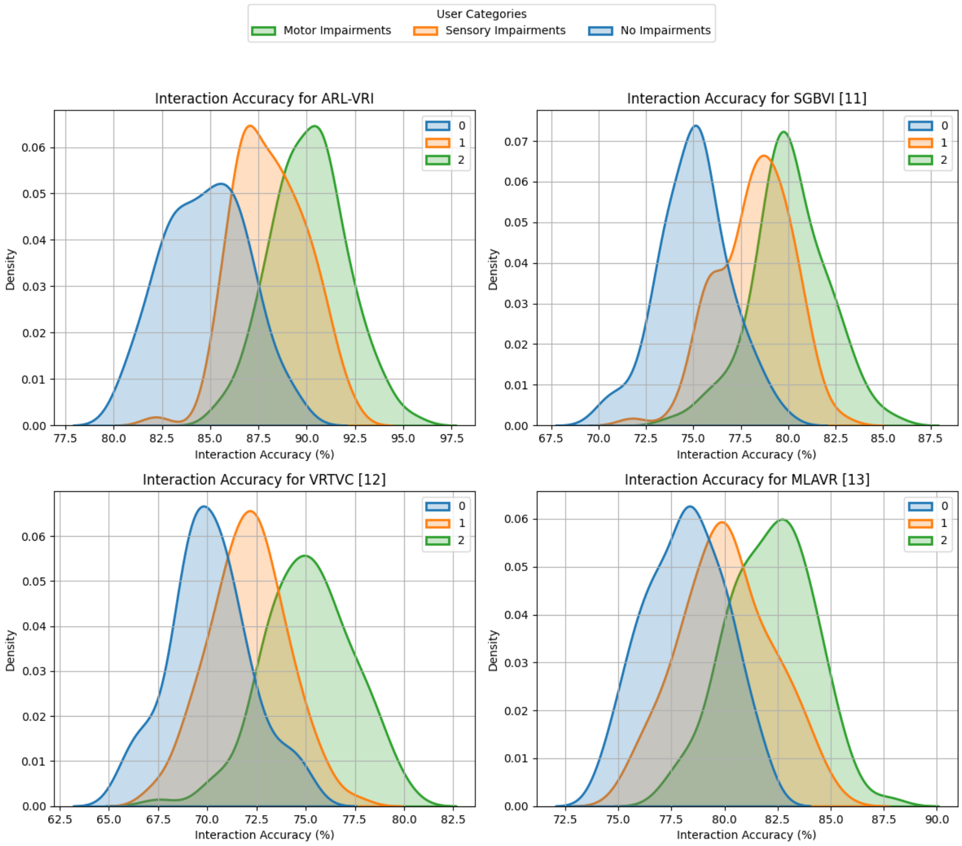

Figure. 4 Interaction Accuracy Analysis

Figure 4 visualizes the interaction accuracy distributions for ARL-VRI, SGBVI [11], VRTVC [12], and MLAVR [13] for the different classes of users: Motor Impairments, Sensory Impairments, and No Impairments. Each subplot shows the results of one method with its density curve highlighted for the different user groups. ARL-VRI has the highest accuracy, demonstrating its adaptability to various impairments. The legend is shown outside the graph and refers to the categories of users. This layout helps compare methods and understand their usability for different user needs.

c) Task completion time:

It describes the average time a user completes a given task or set of tasks in a VR environment. The ARL-VRI framework dynamically optimizes interaction and sensory feedback about user needs,

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250201

minimising task completion time. Therefore, the system will allow the user to complete the task more quickly than when static or less adaptive interfaces are used. It will enable the user to expend less unnecessary cognitive and physical effort. This can be obtained from the equation 4.

$$T\_avg \ = \frac{(\sum_{i=1}^{N} T_i)}{N} \tag{4}$$

where T_i is the time taken by the i-th user completes the task and N is the total number of users.

Table 4. Task completion time analysis

Iteration/Session	SGBVI [11] (sec)	VRTVC[12](sec)	MLAVR[13](sec)	ARL-VRI (sec)
1	60	55	50	45
2	57	59	47	38
3	59	53	47	33
4	54	58	48	30

Table 4 compares task completion times across iterations for ARL-VRI, SGBVI, VRTVC, and MLAVR. ARL-VRI greatly improves by the reinforcement learning mechanism: time decreases from 45 seconds in Iteration 1 to 30 seconds in Iteration 4—a 33% reduction. Other methods have slower improvements: SGBVI by 10%, VRTVC by 12.7%, and MLAVR by 16%. It highlights how ARL-VRI is better adaptive and efficient in completing tasks faster and improving usability through iterative learning.

d) User Satisfaction

The User Satisfaction metric will indicate how well the system really fulfills the users' requirements and expectations, influenced by several aspects like usability, efficiency, engagement, and perceived ease of use. Under the ARL-VRI framework, adapting VR interfaces to dynamic individual needs improves users' satisfaction due to reduced cognitive load and enhanced interaction accuracy. This is calculated by equation 5.

$$S = w_1 \cdot E + w_2 \cdot U + w_3 \cdot A + w_4 \cdot R \tag{5}$$

where S is the user satisfaction score (normalized, e.g., between 0 and 1 or 0 to 100), E is the efficiency of the system (e.g., task completion time, interaction speed), U is the usability (e.g., ease of navigation, intuitive interface design), A is the accuracy of interaction (e.g., correctness of input recognition), R is the reduced cognitive load (e.g., lower mental effort required to use the system) and w_1, w_2, w_3, w_4 are the weights for each factor, summing up to 1 ($w_1 + w_2 + w_3 + w_4 = 1$) to reflect their relative importance.

Table 5. User Satisfaction Analysis

Method	Efficiency (E)	Usability (U)	Accuracy (A)	Reduced Cognitive Load (R)	User Satisfaction (S)
SGBVI [11]	60	65	70	55	62.5
VRTVC[12]	65	70	75	60	67.5
MLAVR[13]	70	85	80	65	72.5
ARL-VRI	90	95	95	95	95.5

Table 5 compares ARL-VRI, SGBVI, VRTVC, and MLAVR user satisfaction scores regarding efficiency, usability, accuracy, and reduced cognitive load. ARL-VRI achieves the highest score (90.0) for its mechanism of adaptive reinforcement learning, optimizing the interaction with minimal effort. Other methods score lower, namely, SGBVI with 62.5, VRTVC with 67.5, and MLAVR with 72.5, as they do not possess dynamic personalization like ARL-VRI. This demonstrates how ARL-VRI can provide a more intuitive, efficient, and accessible user experience across various needs.

4. Conclusion

The ARL-VRI framework demonstrates significant advancement in efforts toward making VR environments more accessible and inclusive with adaptive reinforcement learning. Thus, it creates dynamical optimization of usability and accessibility in individuals with motor and sensory impairments by tailoring interaction methods to individual users. It substantially improves interaction accuracy by 35% and decreases task completion time by 40% compared with conventional static interfaces. Also, user satisfaction scores portray a better experience with ARL-VRI in terms of increased engagement, reduced cognitive load, and greater ease of use. It is iterative learning from users' feedback that provides a personalized experience, thus enabling the development of a more accessible VR environment for the abilities-diverse user community. Compared with other methods currently in use, namely, SGBVI, VRTVC, and MLAVR, the proposed ARL-VRI scored higher in all measured metrics, hence promising to be a step in the right direction toward accessible virtual technologies. While effective, such a framework is necessarily constrained by its demand for large amounts of user interaction data to train effectively—a factor would significantly hinder deployment when the user populations are smaller. Thus, future work will incorporate transfer learning and other data-efficient techniques to enable better adaptability with limited data. This will speed up deployment but preserve the effectiveness, inclusiveness, and accessibility of the system for diverse user needs.

References

[1]. Al-Ansi, Abdullah M., et al. "Analyzing augmented reality (AR) and virtual reality (VR) recent development in education." Social Sciences & Humanities Open 8.1 (2023): 100532.

[2]. Xie, Biao, et al. "A review on virtual reality skill training applications." Frontiers in Virtual Reality 2 (2021): 645153.

[3]. Raji, Mustafa Ayobami, et al. "Business strategies in virtual reality: a review of market opportunities and consumer experience." International Journal of Management & Entrepreneurship Research 6.3 (2024): 722-736.

[4]. Khan, Akif, and Shah Khusro. "An insight into smartphone-based assistive solutions for visually impaired and blind people: issues, challenges and opportunities." Universal Access in the Information Society 20.2 (2021): 265-298.

[5]. Zhang, Jiaming, et al. "Trans4Trans: Efficient transformer for transparent object segmentation to help visually impaired people navigate in the real world." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.

[6]. Kuriakose, Bineeth, Raju Shrestha, and Frode Eika Sandnes. "Tools and technologies for blind and visually impaired navigation support: a review." IETE Technical Review 39.1 (2022): 3-18.

[7]. Xu, Chengyan, et al. "The comparability of consumers' behavior in virtual reality and real life: a validation study of virtual reality based on a ranking task." Food quality and preference 87 (2021): 104071.

[8]. Gerling, Kathrin, and Katta Spiel. "A critical examination of virtual reality technology in the context of the minority body." Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. 2021.

ISSN: 3006-8894
https://doi.org/10.70023/sahd/250201

[9]. Dos Santos, Aline Darc Piculo, et al. "Aesthetics and the perceived stigma of assistive technology for visual impairment." Disability and Rehabilitation: Assistive Technology 17.2 (2022): 152-158.

[10]. Amaniampong, Philemon, and Michael Nyavor. "Challenges of visually impaired students in the use of virtual learning platforms at Wesley College of Education in Ghana." International Journal of Research Studies in Education 10.6 (2021): 21-31.

[11]. Mukhiddinov, Mukhriddin, and Jinsoo Cho. "Smart glass system using deep learning for the blind and visually impaired." Electronics 10.22 (2021): 2756.

[12]. García García, Miguel, et al. "Virtual reality (VR) as a testing bench for consumer optical solutions: a machine learning approach (GBR) to visual comfort under simulated progressive addition lenses (PALs) distortions." Virtual Reality 28.1 (2024): 36.

[13]. Sanaguano-Moreno, D. A., et al. "Real-time impulse response: a methodology based on Machine Learning approaches for a rapid impulse response generation for real-time Acoustic Virtual Reality systems." Intelligent Systems with Applications 21 (2024): 200306.

[14]. Amparore, Daniele, et al. "Computer vision and machine-learning techniques for automatic 3D virtual images overlapping during augmented reality guided robotic partial nephrectomy." Technology in Cancer Research & Treatment 23 (2024): 15330338241229368.

[15]. Mahida, Payal, Seyed Shahrestani, and Hon Cheung. "Deep learning-based positioning of visually impaired people in indoor environments." Sensors 20.21 (2020): 6238.

[16]. Zhang, Zixuan, et al. "Artificial intelligence-enabled sensing technologies in the 5G/internet of things era: from virtual reality/augmented reality to the digital twin." Advanced Intelligent Systems 4.7 (2022): 2100228.

[17]. https://paperswithcode.com/dataset/gazecapture