Autonomous Agents for Accessibility: Simulating Visual Impairments in Web Interfaces

Juan Diego Yepes-Parra
Universidad de los Andes, Colombia
j.yepes@uniandes.edu.co

Camilo Escobar-Velásquez
Universidad de los Andes, Colombia
ca.escobar2434@uniandes.edu.co

***Abstract*—Static code analysis cannot detect real-time interaction issues faced by users with disabilities. We introduce a multimodal AI agent framework that simulates user interactions under visual impairments without code access. These agents are exposed to perceptual filters representing conditions such as glaucoma and myopia, and receive both visual input and audio output generated by screen readers. We outline a framework to evaluate how such filters affect user behavior, task success, and interface usability. Our approach aims to uncover visual accessibility flaws that emerge only through impaired perception. We also reflect on the ethical considerations of simulating user experiences and outline research directions toward agent-based, perception-driven accessibility evaluation.**

***Index Terms*—Web Accessibility; Autonomous Artificial Intelligence (AI) Agents; Automated Testing**

I. Introduction

Web accessibility is central to inclusive software design, yet existing evaluation methods predominantly focus on code conformance, such as Web Content Accessibility Guidelines (WCAG) rule checks, often overlooking how individuals with disabilities actually navigate and perceive interfaces in practice [1]. Although these tools are useful for guidance and a first approach, they can be unable to thoroughly assess how these impairments affect real-time usability, task completion, or perception of interface elements. While often ignoring behavioral context; they do not simulate navigation, focus order, or sequential interaction flows that can significantly impact accessibility. Individuals with disabilities have diverse needs and experiences, many of which may not be fully addressed by level-A WCAG guidelines alone.

Although the access to information is a human right, the urgency of dynamic and human-centered accessibility evaluation is also supported by data. Approximately 2.2 billion people globally have some sort of visual impairment, including both near and distant vision issues [2]. In Colombia, among the estimated 2.65 million people living with a disability by the DANE, approximately 57% report that vision-related activities present the greatest challenges [3]. In bigger countries like the U.S.A., the C.D.C. reports that, about 5.5% of adults (nearly 19 million people) have blindness or serious difficulty seeing [4]. Furthermore, screen reader users find that the most problematic items to interact with in a webpage are CAPTCHA, interactive elements, ambiguous links or buttons, unexpected screen changes, lack of keyboard support, among others [5]. Even more tellingly, these users extract information from data visualizations 61% less accurately and take 211% more time compared to sighted users [6].

We explore the possibility of approaching accessibility testing using autonomous AI agents capable of interacting with web pages visually while being exposed to simulated visual impairments. These agents are prompted with specific user tasks (e.g., locating a button, submitting a form) and attempt to complete them. The agents can also integrate outputs from assistive technologies, for example, capturing a screen reader's textual narration as an additional input channel.

This multi-modal input allows the agent to simulate how a user with various levels of visual impairment (such as glaucoma, cataracts, myopia or low vision, among others) interacts with content, opening the door for dynamic automated testing other existing tools might be unable to uncover. For instance, detecting unuseful alt text or mislabeled controls or mismatches between rendered content and screen reader output, thereby uncovering potential issues that visual or static code checks miss.

This paper outlines our motivation, proposed approach, poses future evaluation research questions, and discusses implementation considerations for such a system.

II. Related Work

A. Web Accessibility compliance

Conventional accessibility assessments primarily rely on static checkers that verify compliance with WCAG criteria by inspecting HTML and CSS structures. Tools such as WAVE by WebAIM and IBM's NPM accessibility-checker exemplify this approach [7]. However, several studies have revealed significant limitations in these tools, pointing out existing tools often lack semantic awareness and fail to consider user perspectives, leading to incomplete or misleading results [1]. For instance, an evaluation of Bulgarian museum websites uncovered widespread accessibility failures despite technical correctness [8]. Similarly, a study of Norwegian municipal websites found that although legislation mandated WCAG compliance, issues such as low-quality alternative text persisted [9].

It is also important to note that WCAG conformance varies by level, from A to AAA, meaning that a website labeled as "compliant" may still fall short of more stringent and specific accessibility standards. These findings underscore the need for real-time evaluations that reflect how visually impaired users interact with web interfaces. While automated tools are useful for early feedback, they are unable to identify all critical accessibility, functionality, and usability issues affecting this population [8].

B. How Artificial Intelligence can be Integrated

Recent work has proposed hybrid testing frameworks that attempt to improve automated evaluation by combining guidelines with heuristic or AI enhanced methods. Machine learning can be used to identify ARIA landmarks in web applications, highlighting the potential of classification models to infer structure when developers omit key accessibility tags [10]. Likewise, various different types of machine learning models can be trained to identify and correct accessibility problems for helping websites follow accessibility guidelines, by looking at the source code and proposing viable solutions [11], [12]. These efforts demonstrate growing awareness of the limitations of rule-based systems, but rely primarily on source code analysis.

C. Autonomous AI agents

The use of autonomous AI agents to simulate user interaction with web interfaces has also been explored. UXAgent is a notable system that uses LLM agents to mimic thousands of diverse user personas in web usability studies. The agents interact with live websites via browser automation, providing qualitative and quantitative feedback that supports iterative UX design [13]. Complementing this, a GitHub Copilot extension that proactively embeds accessibility guidance into the coding workflow was also proposed [14]. Their study shows how AI assistants can suggest accessible UI code, highlight missing attributes, and prompt manual verification during development.

Another recent advancement in this area is AXNav, a system that interprets accessibility test instructions written in natural language and executes them on remote cloud devices using an LLM-based multiagent planner [15]. This approach demonstrates how LLM-driven agents can automate complex accessibility evaluations and provide actionable, context-rich feedback for developers.

Multimodal agents that adaptively present content based on user needs, transforming visual content into speech or simplified visuals for users with auditory or visual processing disorders were also proposed [16]. While not focused on web testing, their work shows how agents can model disability-specific interactions across modalities.

III. Motivation

The widespread adoption of the web has emphasized the importance of ensuring that online content is accessible to everyone, including individuals with various disabilities [17]. Despite this, less than 4% of the top one million websites are fully accessible. Over 96% contain detectable WCAG failures, with an average of 51 errors per page [18]. Common issues include missing alt text (present on 56% of home pages), low contrast, and poor form labeling [19].

Meanwhile, the legal landscape is becoming stricter. The European Accessibility Act mandates compliance by June 28, 2025 for both public and private digital services in the EU, affecting more than 87 million EU residents with disabilities [20], [21].

On the other hand, rigorous testing is integral to ensuring reliability of web applications. For example, usability testing that involves having real users interacting with a live web application, allows for the identification of issues in action, enables for iterative feedback and a more realistic assessment of how they might experience the interface. Effective testing improves satisfaction, avoids rework, and ensures genuinely inclusive design [22].

Although WCAG articulate extensive criteria for inclusive design [23], existing tools focus more on compliance than on realistic user behavior. Developers and UX practitioners often lack practical mechanisms to validate these guidelines in realistic usage scenarios. For instance, some of the significant challenges found in testing are little room for iteration, lack of extensive and appropiate user feedback, and limitations to empathy-based research methods [13].

IV. Automated Testing

Dynamic testing can help developers and stakeholders verify and validate that running software is working as expected. In this case, automated testing is the enabler for faster, more reliable and overall optimized testing. Automated testing leveraged by machine learning and AI are increasingly becoming more popular and sophisticated, therefore using them in this scenario is a big step forward for ensuring accessibility.

Intelligent agents can interact with web content without access to the underlying code, relying instead on the same content the final user is interacting with [13], [24], [25]. They can also be adapted based on feedback, and can process rich multimodal inputs, such as screenshots and screen reader text, enabling them to reason about both the visual layout and the spoken feedback of the user interface simultaneously. They are also able to use different interaction techniques, like keyboard only, mouse only, etc. This integrated perception allows for a more comprehensive assessment of accessibility issues that may arise during real user interactions. Finally, LLM agents excel at open-ended reasoning and can provide qualitative insights, alongside quantitative logs. However, they require careful prompting and can be slower or less predictable.

V. Objectives

First, we aim to explore the use of perceptual filters and multimodal inputs, such as applying blur to simulate different impairments; while also using screen reader outputs, to approximate the experience of real world users.

Additionally, we will analyze whether certain UI elements become inaccessible or difficult to perceive under these simulated conditions, and investigate if common layout structures or design patterns can fail in these situations. Through this approach, we seek to identify not only the direct effects of visual filters on usability, but also the broader structural and behavioral implications for accessible web design.

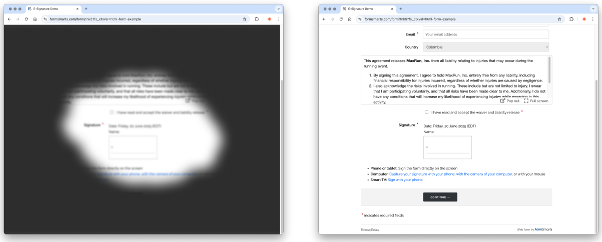

Fig. 1: Left: Glaucoma filter applied to an example form requiring a signature. The "Submit" button is no longer visible, making it difficult to locate. Right: Original version.

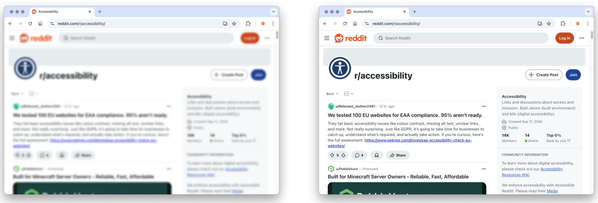

Fig. 2: Left: Myopia filter (-3 diopters) applied to a social media webpage, reducing clarity and edge sharpness. Right: Original version.

VI. System Design

A. Visual Filtering

Simulating visual impairments through perceptual filters allows us to approximate the experiences of users with conditions such as glaucoma and myopia. These filters alter the rendered webpage in ways that reflect known physiological effects, such as peripheral vision loss, blur, or reduced contrast sensitivity.

These simulations (see figures. 1 and 2) were implemented using the Visual Impairments Simulator Chrome extension [26]. While such tools provide a useful approximation of typical visual conditions, they are inherently limited: visual disabilities are highly individualized, and no filter can perfectly replicate the subjective visual experience of every user.

Future work should involve collaboration with ophthalmologists and vision science experts to improve the realism and clinical accuracy of these filters. Their insight can guide the calibration of filter parameters and help us design simulations that more closely resemble the lived experiences of users with specific visual conditions.

B. Assistive output integration

The agent must capture what an assistive technology user would hear and do. For screen reader output, we can integrate a driver API, such as Guidepup that programmatically controls VoiceOver (macOS) or NVDA (Windows). This driver issues the same keyboard commands a user would and exposes the spoken utterances [27]. The agent framework would invoke this API at each step and record the resulting stringof text (including element role, name, state) as sensory input.

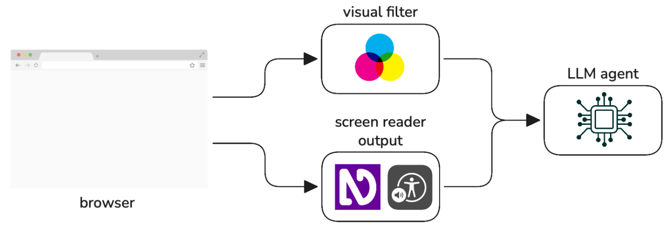

Fig. 3: Proposed pipeline for the agent feeding inputs

For keyboard-only navigation logs, the agent can simply log focus events. As the agent presses Tab, arrow keys, Enter, etc., a browser automation framework, such as Playwright, Selenium or Kraken, can attach listeners to record each focus transition and action. This yields a sequence of elements visited in order [28]. These logs serve two purposes: they ensure the agent only uses keyboard input, and they provide a trace of the navigation path for later analysis and execution. For instance, the log can reveal if an item was skipped or if the focus order was non-standard. In practice, after each key press the agent would pull both the new focused element's DOM attributes and the latest screen-reader text. Together, these multimodal streams visual focus shifts and spoken labels populate the agent's perception at each timestep. This proposed pipeline can be visualized in figure 3.

The agent may optionally inspect the page for accessibility metadata, thereby having the ability to cross-check perception. This means reading the ARIA attributes for the current focused element. For instance, when an element is in focus, the agent can query its `aria-label`, via de Accessibility Object Model or other APIs. This is then compared to the other information (visual, audio). If the visible text of a button says "Search" but its ARIA label is "Submit," that mismatch is noteworthy. Likewise, this is helpful for alt text. If the alt text of an image is `alt= 'Company Logo'` but the screen reader says "Image", it indicates a missing accessible name. This is useful for verifying consistency, which is one of the guidelines on WCAG and IBM's [29].

C. Decision and Action Module

Our approach consists of prompting the agent with a basic user goal, such as "given this screenshot and screen reader transcript, where is the 'Submit' button?" The agent will then attempt to complete the task, using the screen reader and seeing through the visual impairment filter applied to the interface.

For example, one scenario may involve an AI agent attempting to find and click a "Submit" button after filling out a form. With a glaucoma filter applied (see Fig. 1), the button may become difficult to distinguish due to peripheral blur or low contrast.

D. Workflow Definition

Each testing task needs to be defined in advance. This can be done either with a script or using natural language to define a goal. Then, the agent will execute said task in a closed loop that has a completion or failure condition. The loop can be structured in this way

1) Perception: Capture the current UI state. This includes a screenshot of the page and the latest screen-reader output. Optionally also collect DOM/ARIA info for elements in focus.
2) Decision: Use the agent's strategy to choose the next action. An LLM-based agent might generate a plan (e.g. "click the Login button") from the task description, and then a rule-based agent would follow a predetermined script (e.g. "press Tab until username field is focused, then type").
3) Action: Execute the chosen step.
4) Measure: Record relevant outputs. Log the agent's action, the resulting new screen-reader output, and any state changes. Check if the task goal is reached. If not, loop back with the new perception.

E. Metrics and Analysis

After execution is complete, some of the metrics we propose that the system displays are both classic usability and accessibility-specific. These include the agent's task success rate, efficiency (measured by completion time or number of actions), and the frequency of interaction errors such as missed clicks or incorrect actions that require backtracking. We also include the consistency between screen reader output and the visible UI, flagging any mismatches between spoken labels and on-screen text or roles. Visual robustness can also be examined by analyzing the webpage before the filter, discovering layout faults like overlapping or off-screen elements, among others. Heuristic checks are performed during testing, such as verifying that text scales appropriately in high-contrast or large-text modes and monitoring for navigation loops. Throughout the process, we aggregate logs of the agent's actions and screen reader output for offline analysis, enabling manual review or further explanation by the agent. By comparing these metrics across different simulated conditions, we can quantify the impact of visual impairments on usability and identify both layout and semantic accessibility issues.

F. Ethical Considerations

Simulated agents must not be seen as full substitutes for real users with disabilities. They are procies, and never replacements for real user studies. Their effectiveness depends on how accurately they model user behavior, and poor design could lead to misrepresentations of user needs. Eventual human validation is also necessary, to ensure no mismodeling and to take into account privacy considerations.

VII. Evaluation

To investigate the feasibility, effectiveness, and ethical implications of using autonomous AI agents for accessibility evaluation, we pose several research questions. First, we ask whether autonomous AI agents can realistically emulate the interaction patterns of users with vision-related impairments when navigating web interfaces. This includes examining which visual filters or perceptual constraints most effectively represent different types of visual impairments in a simulated context, and what design principles are necessary for agents to approximate impaired visual perception through perception-based (rather than code-aware) interaction.

We also explore whether these agents can surface accessibility problems that static tools overlook, such as poor contrast visibility, confusing focus order, or dynamic content that is not screen-reader friendly. Another important question is whether agent-based testing, when combined with simulated impairments, can produce accessibility assessments that are reliable and generalizable. We are interested in the extent to which LLM-powered agents or learning-based systems can internalize accessibility heuristics through observation of human interaction data, rather than relying solely on static rule sets.

Ethical considerations are central to this research. We examine the risks that arise when relying on simulated agents as stand-ins for real users with disabilities, particularly in terms of misrepresentation or oversimplification. We seek to ensure that agent-based evaluation tools meaningfully contribute to more inclusive web design, rather than substituting for real user feedback, and that agents simulating disabilities preserve the autonomy and dignity of real users, rather than acting as complete surrogates.

Finally, we investigate how combining simulated visual impairment filters with screen reader output affects the agent's ability to detect accessibility issues, and whether multi-modal input can uncover problems, such as missing alt text, that would not surface if using only visual filtering.

Our evaluation plan involves initial experiments using a set of benchmark pages, such as public websites with known accessibility issues. We will define tasks like form submission and navigation, and compare agent results under different filters and input modes. In future work, we may include qualitative validation with accessibility experts or small user studies to further assess the reliability and generalizability of agent-based accessibility evaluations.

VIII. Conclusion & Future Work

This paper proposes a new direction for accessibility evaluation: the use of autonomous, multi-modal AI agents that simulate visual impairments and interact with web content through perception, not code. By combining visual filters, screen reader output, and task-based prompting, this approach aims to approximate the experiences of users with diverse visual disabilities and to surface accessibility issues that may be missed by static tools.

The potential of this method lies in its ability to reveal failures that only emerge under perceptual constraints, and to provide richer, multi-modal insights into web accessibility. At the same time, we recognize the ethical and methodological challenges of using simulated agents as proxies for real users, and stress the importance of human validation.

As next steps, we plan to design and implement a prototype system that integrates visual impairment simulation, screen reader emulation, and agent-based control. We will define representative tasks and benchmark scenarios to explore how different impairments and input modes affect accessibility outcomes. Future work will also involve comparing this approach with existing static tools, and seeking feedback from accessibility experts and users. Ultimately, we hope this research will lay the groundwork for more dynamic, user-centered accessibility evaluation methods and inspire further investigation in this area.

REFERENCES

[1] J. Ara, C. Sik-Lanyi, A. Kelemen, and T. Guzsvinecz, "An inclusive framework for automated web content accessibility evaluation," *Universal Access in the Information Society*, pp. 1–27, 2024.

[2] World Health Organization, "World report on vision," https://www.who.int, 2023.

[3] Departamento Administrativo Nacional de Estadística (DANE), "Estado actual de la medición de la discapacidad en colombia," 2022, pDF Presentation. [Online]. Available: https://www.dane.gov.co/files/investigaciones/notas-estadisticas/abr_2022_nota_estadistica_Estado%20actual_de_la_medicion_de_discapacidad_en%20Colombia_presentacion.pdf

[4] U. C. for Disease Control and Prevention, "Disability impacts all of us infographic," https://www.cdc.gov/disability-and-health/articles-documents/disability-impacts-all-of-us-infographic.html, 2025.

[5] WebAIM. (2025) Webaim: Screen reader survey results. Accessed: 2025-06-24. [Online]. Available: https://webaim.org/projects/screenreader/

[6] J. O. Wobbrock *et al.*, "Understanding screen-reader users' experiences with online data visualizations," in *ASSETS '21: The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, 2021, p. 16.

[7] W3C Web Accessibility Initiative, "Accessibility evaluation tools list," 2025, accessed: 2025-06-24. [Online]. Available: https://www.w3.org/WAI/test-eval/tools/

[8] T. Todorov, G. Bogdanova, and M. Todorova-Ekmekci, "Accessibility of bulgarian regional museums websites," *International Journal of Advanced Computer Science and Applications*, vol. 13, no. 3, 2022.

[9] Y. Inal and A. B. Torkildsby, "Does the law make a difference? a longitudinal study on accessibility compliance of norwegian municipality websites," *Universal Access in the Information Society*, pp. 1–11, 2025.

[10] W. M. Watanabe, G. de Lemos, and R. W. Nascimento, "Accessibility landmarks identification in web applications based on dom elements classification," *Universal Access in the Information Society*, vol. 23, no. 2, pp. 765–777, 2024.

[11] V. Ramineni, B. S. Ingole, A. R. Banarse, M. S. Krishnappa, N. K. Pulipeta, and V. Jayaram, "Leveraging ai and machine learning to address ada non-compliance in web applications: A novel approach to enhancing accessibility," in *2024 Third International Conference on Artificial Intelligence, Computational Electronics and Communication System (AICECS)*. IEEE, 2024, pp. 1–6.

[12] K. Kuszczyński and M. Walkowski, "Comparative analysis of open-source tools for conducting static code analysis," *Sensors*, vol. 23, no. 18, p. 7978, 2023.

[13] Y. Lu, B. Yao, H. Gu, J. Huang, J. Wang, Y. Li, J. Gesi, Q. He, T. J.-J. Li, and D. Wang, "Uxagent: A system for simulating usability testing of web design with llm agents," *arXiv preprint arXiv:2504.09407*, 2025.

[14] P. Mowar, Y.-H. Peng, J. Wu, A. Steinfeld, and J. P. Bigham, "Codea11y: Making ai coding assistants useful for accessible web development," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 2025, pp. 1–15.

[15] M. Taeb, A. Swearngin, E. Schoop, R. Cheng, Y. Jiang, and J. Nichols, "Axnav: Replaying accessibility tests from natural language," in *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*, 2024, pp. 1–16.

[16] A. Rajagopal, V. Nirmala, I. J. Jebadurai, A. M. Vedamanickam, and P. U. Kumar, "Design of generative multimodal ai agents to enable persons with learning disability." in *ICMI Companion*, 2023, pp. 259–271.

[17] I. Abu Doush, K. Sultan, M. A. Al-Betar, Z. Almeraj, Z. A. A. Alyasseri, and M. A. Awadallah, "Web accessibility automatic evaluation tools: to what extent can they be automated?" *CCF Transactions on Pervasive Computing and Interaction*, vol. 5, no. 3, pp. 288–320, 2023.

[18] WebAIM. (2025) Webaim: The webaim million. Accessed: 2025-06-24. [Online]. Available: https://webaim.org/projects/million/

[19] AudioEye, "Web accessibility stats and data 2024," https://www.audioeye.com/post/accessibility-statistics/, 2024.

[20] ——, "European accessibility act: Ultimate guide," https://www.audioeye.com/post/eu-digital-accessibility/, 2025.

[21] A. Centre, "The eaa comes into effect in june 2025," https://accessible-eu-centre.ec.europa.eu/content-corner/news/eaa-comes-effect-june-2025-are-you-ready-2025-01-31_en, 2025.

[22] A. Studio, "Beyond resolutions: Why 2025 is the year to invest in website accessibility," https://accessdesignstudio.com/beyond-resolutions-why-2025-is-the-year-to-invest-in-website-accessibility/, 2025.

[23] W3C, "Web content accessibility guidelines (wcag) 2.2," 2023. [Online]. Available: https://www.w3.org/TR/WCAG22/

[24] M. Lanham, "Getting started with ai agents," in *AI Agents in Action*. Shelter Island, NY: Manning Publications, 2025, ch. 1.

[25] L. Wang, C. Ma, X. Feng, Z. Zhang, H. Yang, J. Zhang, Z. Chen, J. Tang, X. Chen, Y. Lin *et al.*, "A survey on large language model based autonomous agents," *Frontiers of Computer Science*, vol. 18, no. 6, p. 186345, 2024.

[26] airadavometra, "Simulator of visual impairments," https://chromewebstore.google.com/detail/simulator-of-visual-impai/imohhjdbajiihdogpnhpgbocfodbop, 2023, chrome Extension, Version 1.0.1, accessed: 2025-06-26.

[27] Guidepup Team, "Guidepup: Screen reader driver for test automation," https://www.guidepup.dev, 2025.

[28] W. Ravelo-Méndez, C. Escobar-Velásquez, and M. Linares-Vásquez, "Kraken 2.0: A platform-agnostic and cross-device interaction testing tool," *Science of Computer Programming*, vol. 225, p. 102897, 2023.

[29] IBM Corporation, "Ibm accessibility: Equal access toolkit," https://www.ibm.com/able/toolkit/verify, 2025.