ShadAR: LLM-driven shader generation to transform visual perception in Augmented Reality

Yanni Mei
TU Darmstadt
Darmstadt, Germany
yanni.mei@tu-darmstadt.de

Samuel Wendt
TU Darmstadt
Darmstadt, Germany
samuel.wendt@stud.tu-darmstadt.de

Jonas Wombacher
TU Darmstadt
Darmstadt, Germany
jonas.wombacher@tu-darmstadt.de

Florian Müller
TU Darmstadt
Darmstadt, Germany
florian.mueller@tu-darmstadt.de

Jan Gugenheimer
TU Darmstadt
Darmstadt, Germany
jan.gugenheimer@tu-darmstadt.de

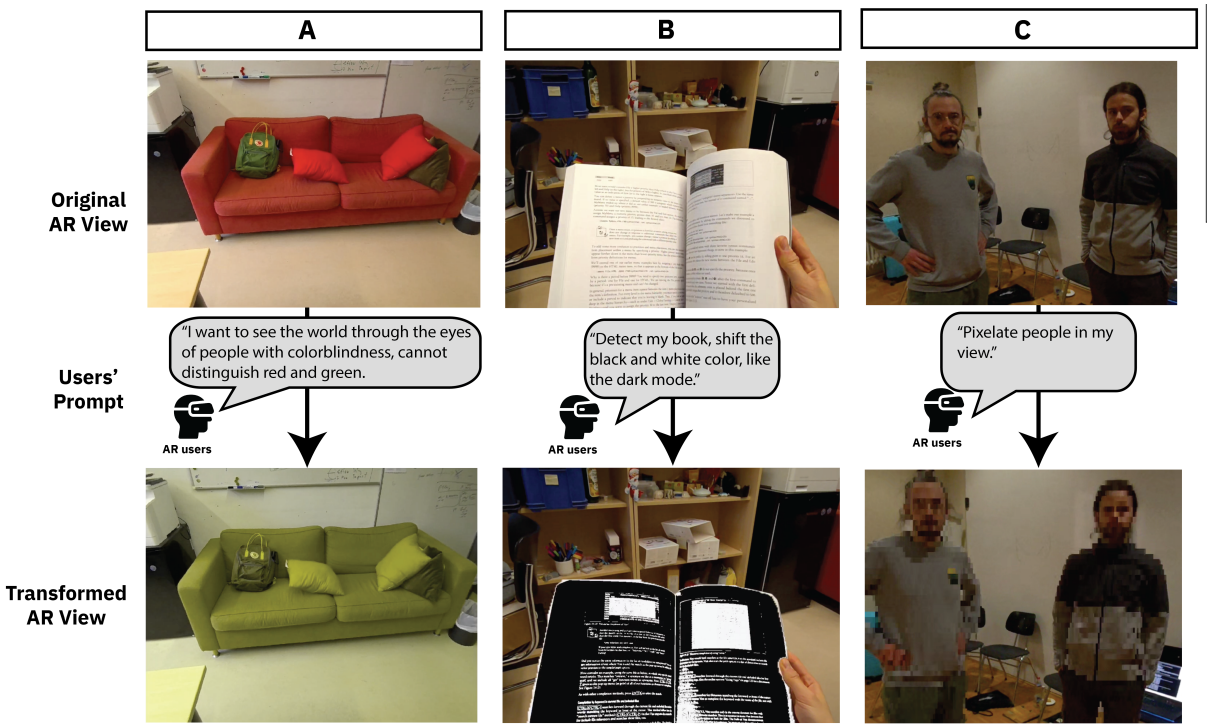

Figure 1: ShadAR enables AR users to transform their view in real time through natural language prompts, generating and applying custom HLSL shaders to AR viewport instantly. Examples include: (A) simulating red–green color blindness, (B) adding dark-mode visual effects to a book, and (C) pixelating people in the surrounding environment.

Abstract

Augmented Reality (AR) can visually transform a user's world by rendering virtual content on top of reality. However, developing such AR apps and visualizations remains a complex process that requires an understanding of computer vision and programming skills. We present ShadAR, an AR prototyping pipeline that enables real-time creation of small AR visualizations and applications using large language models (LLMs) and object detection. ShadAR allows users to express their visual intent (e.g., pixelate every person around me) via natural language, which is interpreted by an LLM to generate corresponding shader code. This shader is then compiled in real-time and applied to the passthrough video stream.

CHI EA '26, Barcelona, Spain

ACM ISBN 979-8-4007-2281-3/26/04
https://doi.org/10.1145/3772363.3799378

CCS Concepts

• **Human-centered computing** → **Mixed / augmented reality**.

Keywords

Augmented reality, Visual augmentation, Large language models, Shader generation

ACM Reference Format:

Yanni Mei, Samuel Wendt, Jonas Wombacher, Florian Müller, and Jan Gugenheimer. 2026. ShadAR: LLM-driven shader generation to transform visual perception in Augmented Reality. In *Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems (CHI EA '26), April 13–17, 2026, Barcelona, Spain.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/3772363.3799378

1 Introduction

Augmented Reality (AR) has opened up new possibilities for directly modifying our visual perception, from simulating how others see the world, to creating entirely novel visual experiences for artistic expressions[7]. This capability has been explored in diverse domains. For example, it allows designers to understand the world as people with different visual capabilities (e.g, cataract) for more inclusive design decisions [1, 6, 10]. It has also been used by artists to creatively stylize the environment, as perceived through the imagined eyes of Van Gogh[3]. However, current implementations of such visual perceptual simulations are typically created beforehand by the developer, designed for a single visual condition, and thus, lack flexibility for broader exploration.

Recent advances in large language models (LLMs) have demonstrated their ability to generate functional code from natural language instructions[4]. Thus, LLMs have been adopted in extended reality (XR) environments to support real-time XR content creation [9]. For example, De La Torre et al. [2] demonstrated how to enable users to generate or manipulate virtual objects in extended reality(XR) with voice commands, using LLM to real-time generate CSharp code.

While prior work has centered on generating source code to create/manipulate XR objects, we take a different approach: using LLMs to generate HLSL shaders for creating/changing AR visual experiences and allowing user to create small AR visualizations and apps. In this project, we present ShadAR, a LLM-enabled shader generation pipeline for AR, deployed on the Meta Quest 3. ShadAR enables AR users to verbally express their desired visual experience (e.g., "I want to see the world through the eyes of someone who is colorblind." or "I want blur every person in my enviornment"). Then, a *ShaderGeneratorLLM*, guided by designed prompts, interprets the user's spoken command and generates HLSL shader code in real time. Additionally, we are running an object detection and segmentation model (Yolo11) on the video stream, which generates a set of labels and coordinates that are streamed into the unity project. In our system prompt, we instructed the LLM on how it can access this data in its code, enabling it to write shaders that can be applied to a specific region (e.g., pixelate every person). The shader is then compiled and immediately applied to the headset's camera passthrough, transforming the user's visual experience on the fly.

We show example applications to demonstrate ShadAR's ability to flexibly interpret user intent and generate corresponding visual effects — though not always perceptually accurate, these effects allow for rapid exploration of altered visual perceptions.

We previously presented an early version of this interactivity at ISMAR 2025 [8]. Since the initial presentation, we have substantially improved the experience by adding an object detection and segmentation model to the pipeline. This addition enables more complex applications that can provide a closer integration with the physical world. For example, this new pipeline enables the system to generate shaders that provide selective manipulations (e.g., pixelating people) or reactions to events in the physical world (e.g., highlighting approaching cars).

2 Technical Architecture

To enable real-time visual transformations from natural language input, ShadAR integrates a pipeline (Fig.2) including four parts:

Intent Understanding and Shader Generation. User voice commands are transcribed using Whisper and sent via HTTP to a prompt-engineered LLM (OpenAI `gpt-5.2`). The prompt incorporates domain knowledge (e.g., animal vision effects), example shader templates, and common pitfalls in shader programming to enhance output reliability.

Object Segmentation We integrated `yolo11-nano` [5] as an object segmentation model in the pipeline. The Yolo model is running on a laptop which receives the current frame of the video passthrough and then populates a list containing object label, the position and outline of the detected object as a set of points.

Real-time Shader Compilation and Deployment. The generated shader code is compiled on a laptop using Unity in headless batch mode, packaged into an asset bundle, then deployed to the Quest device via a local web server.

Accessing and Transforming the AR Viewport. We utilize the *Meta Passthrough Camera API* to access and change the AR headset's video stream as a *WebCamTexture* in *Unity3D*.

3 Demo Experience

Verbally Prompt Visual Perceptions. To experience ShadAR, the user needs to wear the Meta Quest 3 headset, and speak aloud the visual effect they want to experience—such as saying, *"I want to see the world through the eyes of a color-blind person."* While describing their desired effect, they press and hold the Oculus B button to record their voice. After approximately 1 minute, a custom shader is generated and applied to the video passthrough in the user's AR viewport. Users can choose to save the shader for future use, or record a new voice command to generate new visual effect. User's are able to also iterrate on their created shader (e.g., "change the blue highlight to green").

Monoscopic Simulates Stereoscopic. In our setup, the AR viewport is rendered on a 2D quad displaying the left-eye camera feed, resulting in a monoscopic view. Then the compiled shader is applied to the material of the quad rendering the passthrough video, enabling real-time visual perception transformation.

Walkthrough A user that experiences the demo will be instructed to test three specific tasks: 1) Experience the world through the eyes of someone else (e.g., "I want to see the world through the eyes of a dog") 2) Augmenting specific objects in the enviornment (e.g.,"Pixelate every Person around me", "Make every plant glow") 3) Ask for an augmentation that tries to change the mood of the user (e.g., "Make me laugh", "Scare me"). We will bring multiple objects

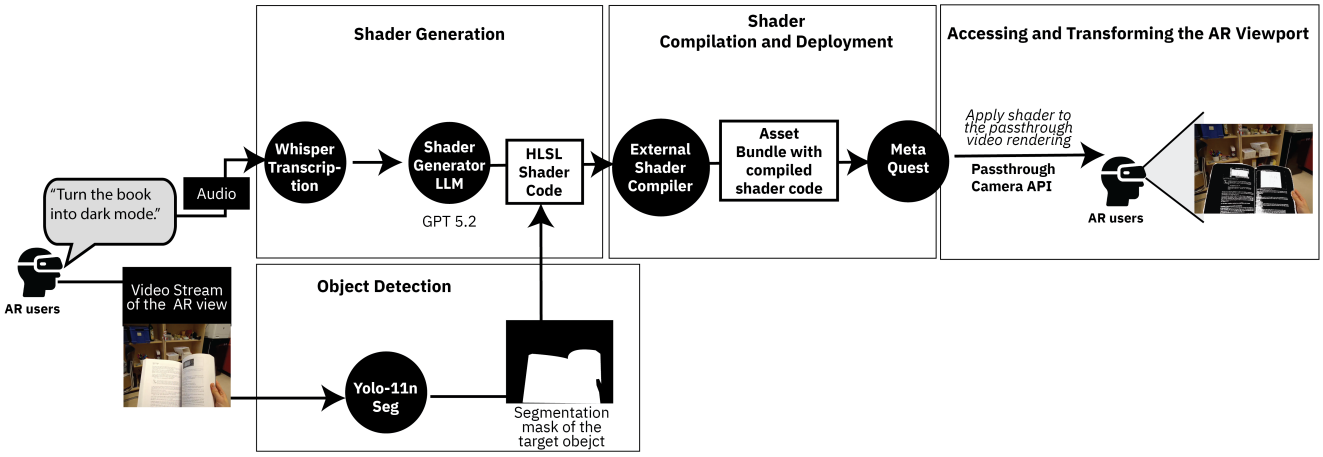

Figure 2: Technical architecture of ShadAR

which are in the Yolo model (e.g., bottle, vegetables) to allow users to test small interactive AR applications which they can generate with the shader (e.g., "Create a small explosion animation when the bottle touches the table"). The full demo experience takes around 5 minutes per person. We plan to bring multiple devices so that multiple persons can experience the demo simultaneously.

4 Example Visualizations and Applications

We illustrate selected examples of using shader generation for visual perception transformation. These examples span a range of goals:

Accessibility Simulation of colorblindness (*"I want to see the world through the eyes of a colorblind person."*, Fig.1-A) or other visual impairments can help designers understand the challenges faced by people with different visual capabilities, enabling more inclusive design choices. Additionally, these shaders can be used to address some specific accessibility issues (e.g., "I am a colorblind person, modify the image so I can recognize green and blue better.").

Creative Expression Creating effects like viewing the world from under the ocean (Fig.3-A) or through a kaleidoscope (Fig.3-D) showcases how such transformations enable artists to explore unique aesthetics and convey specific narratives.

Color Highlighting Prompts like *"I want to see things in grayscale except green"* make it easier for users to find the green folder in cluttered environments (Fig. 3-B). It illustrates how visual perception transformations can support navigation tasks in productivity scenarios.

Object Augmentation Through the integration of the object segmentation model, the system allows users to augment, highlight or dim specific objects in the environment. This can be used for a variety of purposes, such as directing attention (e.g., *"Blur everything except my display."*) (Fig. 3-C) or helping find specific objects (*"Among banana, broccoli and cake, highlight the food with highest calorie."*) (Fig. 3-F).

Interactive Applications The setup also allows for generating simple interactive experiences (e.g., *"Create a wave animation on the table when I place a bottle on it."*) (Fig. 3-E).

5 Conclusion

In this work, we present ShadAR, an AR application that leverages LLMs to interpret users' verbal commands and generate corresponding shader code, enabling real-time transformation of visual perception. We explain the system architecture and showcase examples of generated visual experiences that support inclusiveness, creativity, and searching. We aim to make AR a more flexible "visual experience machine", utilizing the code generation capability of LLMs.

Acknowledgments

This work has been funded by the German Federal Ministry of Education and Research and the Hessian Ministry of Higher Education, Research, Science and the Arts within their joint support of the National Research Center for Applied Cybersecurity ATHENE and by the LOEWE initiative (Hesse, Germany) within the emergenCITY center [LOEWE/1/12/519/03/05.001(0016)/72].

References

[1] Halim Cagri Ates, Alexander Fiannaca, and Eelke Folmer. 2015. Immersive simulation of visual impairments using a wearable see-through display. In *Proceedings of the ninth international conference on tangible, embedded, and embodied interaction*. 225–228.

[2] Fernanda De La Torre, Cathy Mengying Fang, Han Huang, Andrzej Banburski-Fahey, Judith Amores Fernandez, and Jaron Lanier. 2024. Llmr: Real-time prompting of interactive worlds using large language models. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–22.

[3] Vladimir Geroimenko. 2014. Augmented reality art. *Plymouth: Springer* (2014).

[4] Juyong Jiang, Fan Wang, Jiasi Shen, Sungju Kim, and Sunghun Kim. 2024. A survey on large language models for code generation. *arXiv preprint arXiv:2406.00515* (2024).

[5] Rahima Khanam and Muhammad Hussain. 2024. YOLOv11: An Overview of the Key Architectural Enhancements. arXiv:2410.17725 [cs] doi:10.48550/arXiv.2410.17725

[6] Katharina Krösl, Carmine Elvezio, Laura R Luidolt, Matthias Hürbe, Sonja Karst, Steven Feiner, and Michael Wimmer. 2020. CatARact: Simulating cataracts in augmented reality. In *2020 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 682–693.

[7] Tobias Langlotz, Jonathan Sutton, and Holger Regenbrecht. 2024. A Design Space for Vision Augmentations and Augmented Human Perception using Digital Eyewear. In *Proceedings of the 2024 CHI Conference on Human Factors in Computing Systems*. 1–16.

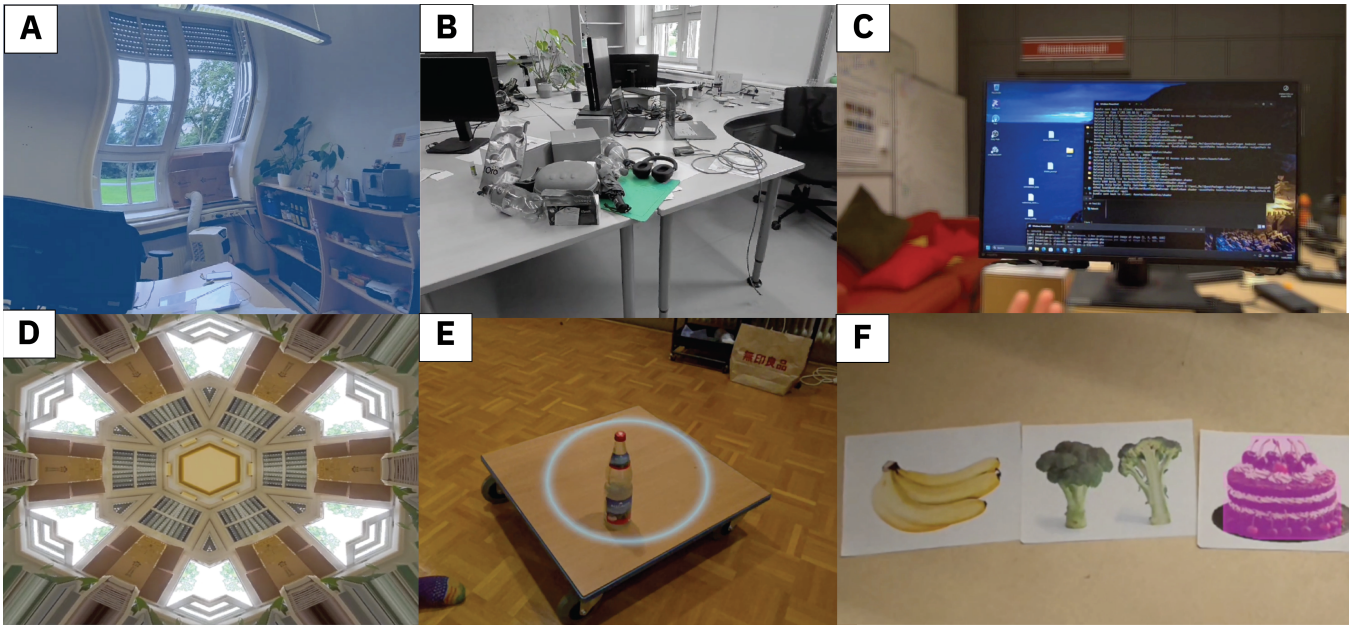

Figure 3: Examples of the generated shaders include: (A) creating an underwater-like visual effect, (B) highlighting green objects to target a green filter within a cluster, (C) blurring everything except the monitor to enhance concentration, (D) generating kaleidoscope-style visual effects, (E) creating an animated ripple-wave effect on a table when a bottle is placed on it, and (F) highlighting the food item with the highest calorie content among a banana, broccoli, and cake.

[8] Yanni Mei, Samuel Wendt, Florian Müller, and Jan Gugenheimer. 2025. ShadAR: LLM-driven Shader Generation to Transform Visual Perception in Augmented Reality. In *2025 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*. 959–960. doi:10.1109/ISMAR-Adjunct68609.2025.00267

[9] Yiliu Tang, Jason Situ, Andrea Yaoyun Cui, Mengke Wu, and Yun Huang. 2025. LLM Integration in Extended Reality: A Comprehensive Review of Current Trends, Challenges, and Future Perspectives. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*. 1–24.

[10] Qing Zhang, Giulia Barbareschi, Yifei Huang, Juling Li, Yun Suen Pai, Jamie Ward, and Kai Kunze. 2022. Seeing our blind spots: smart glasses-based simulation to increase design students' awareness of visual impairment. In *Proceedings of the 35th Annual ACM Symposium on User Interface Software and Technology*. 1–14.