Hyperion: A 3D Visualization Platform for Optical Design of Folded Systems

Daniel K. Nikolov, Sifan Ye, Sydney Dlhopolsky, Zhen Bai, Yuhao Zhu, Jannick P. Rolland (University of Rochester)

***Abstract*—Hyperion is a 3D visualization platform for optical design. It provides a fully immersive, intuitive, and interactive 3D user experience by leveraging existing AR/VR technologies. It enables the visualization of models of folded freeform optical systems in a dynamic 3D environment. The frontend user experience is supported by the computational ray-tracing engine of Eikonal+, an optical design research software currently being developed. We have built a cross-platform light-weight version of Eikonal+ that can communicate with any user interface or other scientific software. We have also demonstrated a prototype of the Hyperion 3D user experience using a Hololens AR display.**

Keywords—Eikonal+, 3D AR/VR scientific visualization, optical design, freeform optics, human interaction, Hololens

I. INTRODUCTION

Conventional optical systems consist in large part of a sequence of linearly positioned lenses, which share a 2D cross section and are inherently rotational symmetric ((1) in Fig. 1). As a result, today's optical system design software provides a 2D interface to designers through a monitor, a mouse, and a keyboard. However, emerging optical technologies that are key to AR/VR applications, such as freeform optics, add many more levels of freedom and complexities to the optical system design, challenging the conventional design software [1, 2].

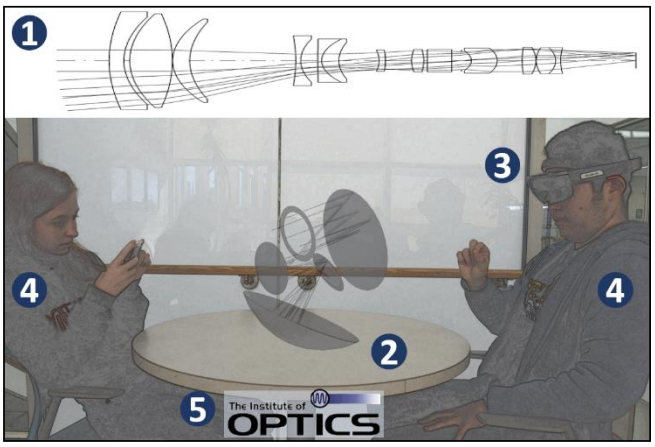

Fig. 1.Project motivation and vision: 1) conventional in-line geometry; 2) freeform folded geometry; 3) AR/VR display for a true 3D user experience; 4) collaborative design process; 5) training and educational applications.

Visualizing and modeling these geometries in the conventional 2D fashion is challenging for the designers. The visualization platform Hyperion addresses these challenges by providing AR/VR enabled ((3) in Fig. 1) interactive 3D user experience, collaborative design capabilities ((4) in Fig. 1) and a tool for education and training of beginner optical engineers.

II. IMPLEMENTATION

The backend is based on the legacy Eikonal ray tracing software that has been modified [3]. It is responsible for carrying out the fundamental computational functionality including ray tracing. A separate abstraction layer has been developed in Python to assist the communication with the backend. The Hyperion frontend has been built in the Unity graphics rendering engine due to its cross-platform capabilities.

Existing literature shows that considering the user's cognition and perception is essential when developing AR/VR visualization tools like Hyperion [4]. The implementation conforms to some existing VR/AR design principles and guidelines, including low short-term memory load, keeping users in control, and depth budgeting. Ongoing work includes applying existing VR/AR usability guidelines to the Hyperion development. The implementation should have more user feedback, include informative dialogs, and reduce free-space interaction to avoid "Gorilla Arm".

The initial prototype demonstrates the visualization of a test optical system. A user can interact with the system with pinch gesture by holding and moving an optical surface and the ray-tracing visualization will be updated in real-time. The ray and system data are updated using the completed backend-to-frontend communication pipeline. The user can also reposition the optical system in 3D space by pinching and dragging the system anchor. Further functionality is currently being developed including tilting, adding and removing of optical surfaces, as well as improving the usability.

REFERENCES

[1] Forbes, Shape specification for axially symmetric optical surfaces. Optics Express 15, 5218 (2007).

[2] Cakmakci, O., Moore, B., Foroosh, H. & Rolland, J. Optimal local shape description for rotationally non-symmetric optical surface design and analysis. Optics Express 16, 1583 (2008).

[3] Rayces, JL & Lebich, L. Ray-Code: An Aberration Coefficient Oriented Lens Design And Optimization Program. Proc. SPIE 0766, Recent Trends in Optical Systems Design and Computer Lens Design Workshop 230 (1987). doi:10.1117/12.940229

[4] Sherman, William R. [VNV]. Understanding Virtual Reality: Interface, Application, and Design. Elsevier, 2002.

[5] Kim, Taek-Bong. "Graphical Menus for Virtual Environments." Institute for Graphical Interfaces, July 2005.