

INTRODUCTION

Virtual Reality (VR) is a promising future. It sets up a virtual environment providing an (audio)visual user experience by creating an artificial world where a user is directly placed into. Extensible by haptic systems, it allows to simulate lifelike experiences to any user. Over the last years, VR hardware has more and more turned into something that is affordable and thus available to common end-users in daily life. At the point of time only limited by physical restrictions such as the walls of a room or the cables connecting the hardware devices to a computer, a user is able to move around inside that VR, i.e. physical movement is directly applied onto the artificially generated world. To experience VR, one is usually equipped with a so-called Head Mounted Device (HMD). Containing a screen in front of each eye, this device renders whatever wanted in front of the users eyes. Today's VR products are used in different fields such as training simulations, gaming and even videos. This work presents how VR is integrated into medical applications by developing a VR plugin for a medical image processing framework called MeVisLab. The thread-based plugin has been developed using OpenVR, a VR library that can be used for developing vendor and platform independent VR applications and tested with the HTC Vive.

This work presents how VR is integrated into medical applications by developing a VR plugin for a medical image processing framework called MeVisLab. The thread-based plugin is tested using the HTC Vive (Figure 1).



Fig. 1 An examination of medical data in VR.

METHODS

Data – Several high-resolution Computed Tomography (CT) scans from real patients have been used to develop, test and evaluate the VR plugin.

Workflow – The overall goal of the HTC-Vive MeVisLab plugin is to connect MeVisLab to Virtual Reality. It does so by establishing a connection between those two (Figure 2).

Implementation – The plugin consists of three parts (Figure 3). The first part wraps the functionality of the actual application into a module that can be used by MeVisLab. The second part renders the medical data. The third part forwards the rendered scene to Virtual Reality using OpenVR.

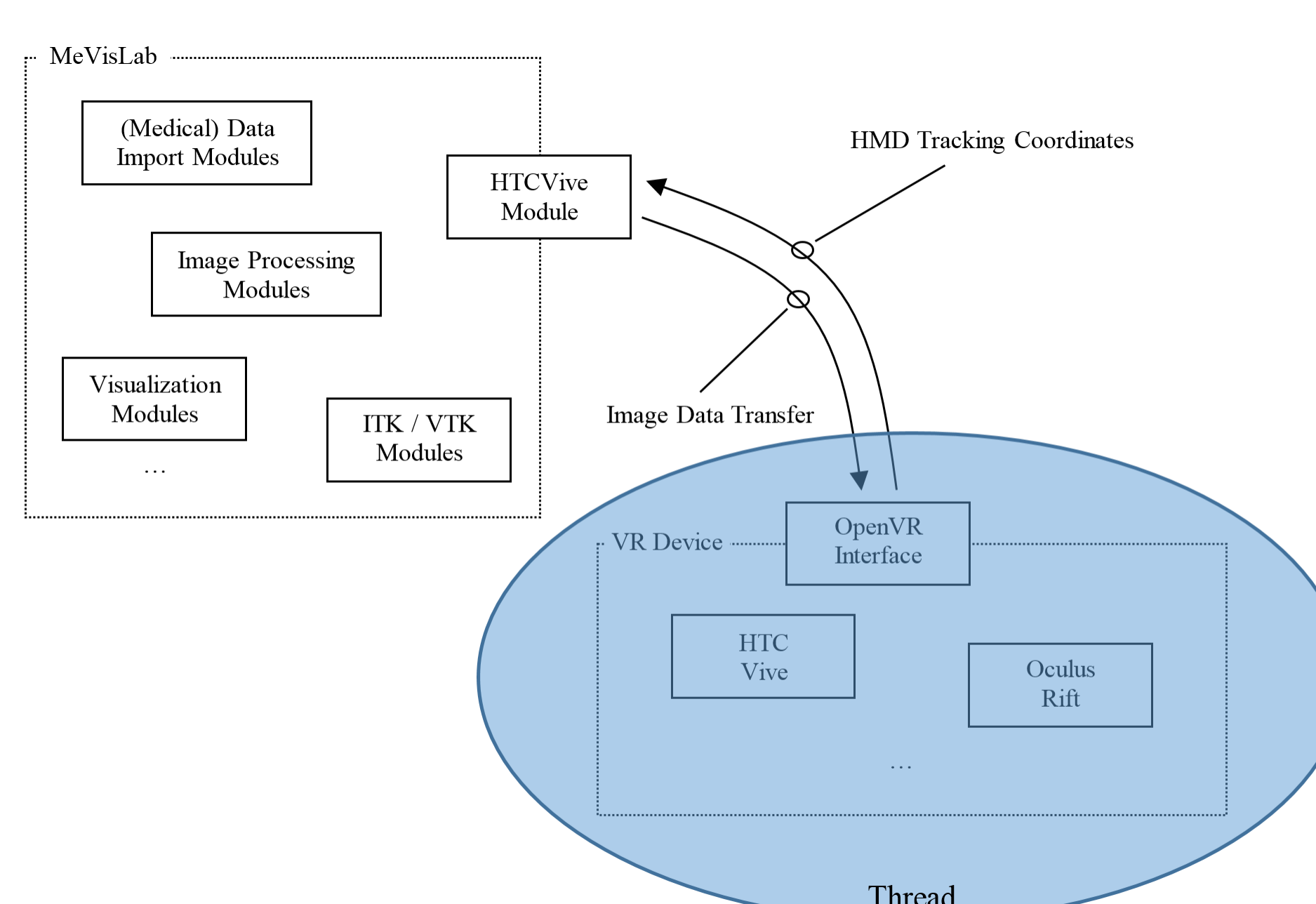


Fig. 2 High level workflow diagram showing the communication and interaction between MeVisLab and the HTC Vive via OpenVR capsuled by an own thread.

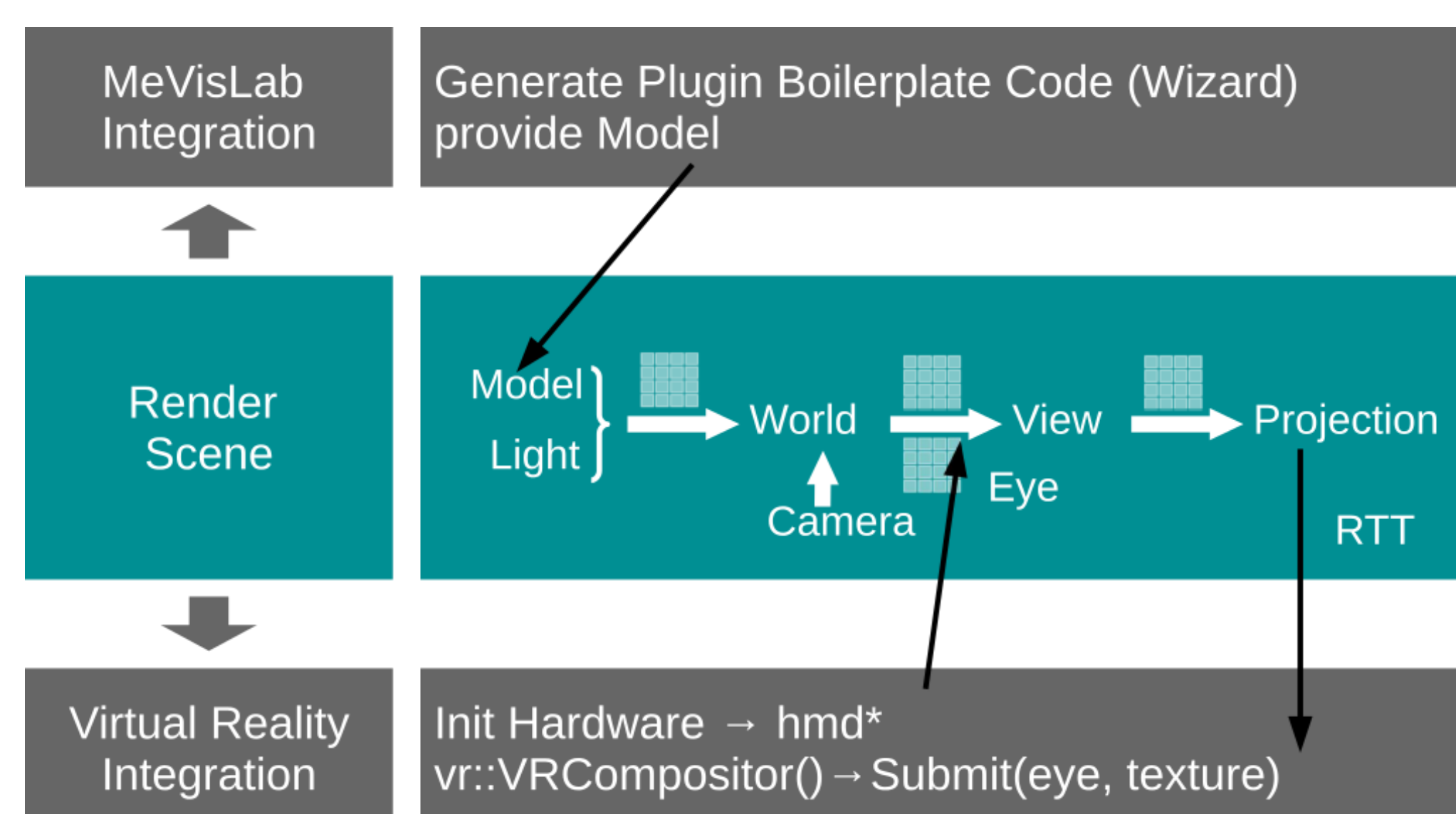


Fig. 3 Diagram showing the three parts of our module implementation and their interaction with each other.

Using OpenVR allows the plugin to work with different VR systems. The plugin can be integrated into any network created by MeVisLab and can thus be a good visual support whenever it comes to the inspection of processed 3D medical data.

RESULTS

The goal of this work was to show the feasibility of an enhanced thread-based integration of Virtual Reality into MeVisLab.

The plugin has been developed and tested under Microsoft Windows 8.1 with the MeVisLab 2.8.1 (21-06-2016) Version for Windows Visual Studio 2015 X64 and OpenVR SDK 1.0.2 (Figure 4).

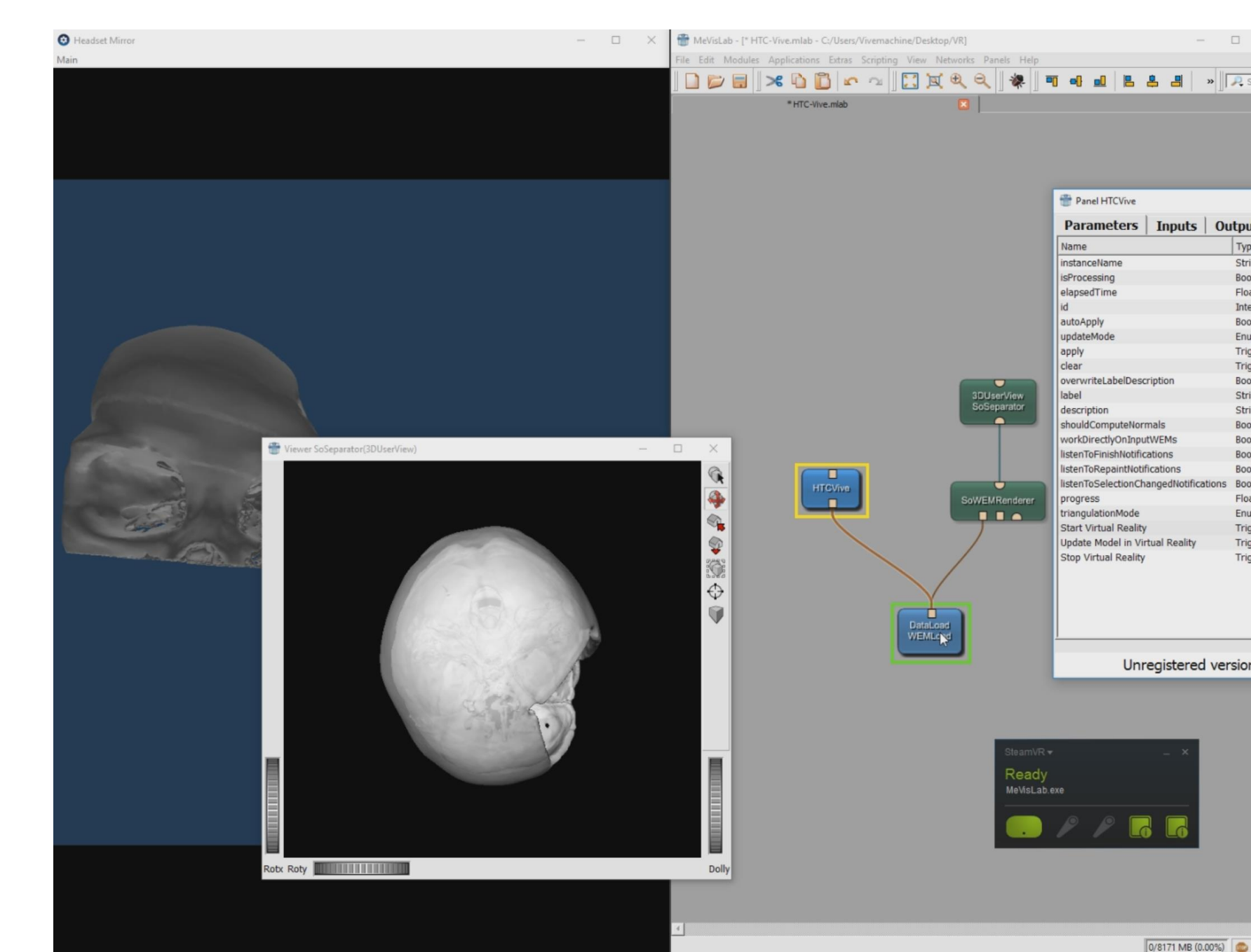


Fig. 4 A MeVisLab sample network using the plugin to render the given 3D medical data in VR.

CONCLUSIONS

1. OpenVR has been integrated into MeVisLab allowing MeVisLab to visualize data in Virtual Reality;
2. A threaded solution enables the simultaneous use of MeVisLab while a user explores the interior of a real human body in Virtual Reality;
3. The integration has successfully been tested with the HTC Vive device;
4. The plugin can be used in different clinical applications and provides a good visual support when inspecting 3D medical data.

Future work includes to allow modifications done in MeVisLab to be rendered to VR in real-time as well as the interaction with the 3D data in VR.

REFERENCES

1. Gall M., et al. "Integration of the HTC Vive into the medical platform MeVisLab," SPIE Medical Imaging (2017).
2. Gall, M., Li, X., Chen, X., Schmalstieg, D. & Egger, J. "Cranial Defect Datasets," ResearchGate (2016).
3. Egger, J. et al. "HTC Vive MeVisLab integration via OpenVR for medical applications," PLoS ONE 12(3): e0173972 (2017).