A comprehensive Workflow and Framework for immersive Virtual Endoscopy of dissected Aortae from CTA data

Jan Egger [a,b,c,d], Simon Gunacker [a,d], Antonio Pepe [a,d,e], Gian Marco Melito [f]
Christina Gsaxner [a,b,d], Jianning Li [a,d], Katrin Ellermann [f], Xiaojun Chen [c]

[a] Graz University of Technology, Institute of Computer Graphics and Vision, Graz, Austria
[b] Medical University of Graz, Department of Oral and Maxillofacial Surgery, Graz, Austria
[c] Shanghai Jiao Tong University, School of Mechanical Engineering, Shanghai, China
[d] Computer Algorithms for Medicine Laboratory, Graz, Austria
[e] Stanford University, School of Medicine, Department of Radiology, Stanford, California, USA
[f] Graz University of Technology, Institute of Mechanics, Graz, Austria

Introduction

Given underlying computer simulations, this work provides a way to visualize Aortic Dissection (AD) in a realistic manner. Based on Virtual Reality (VR), the solution provides a visual user experience that is immersive: while conservative methods might require a researcher to sight a huge amount of medical images, VR allows the researcher to examine the problem from the inside. The presented approach is based on real-patient Computed Tomography (CT) scans. Given these, the aorta is segmented. In a next step, the segmented aorta is converted into a 3D model that can be examined in VR.

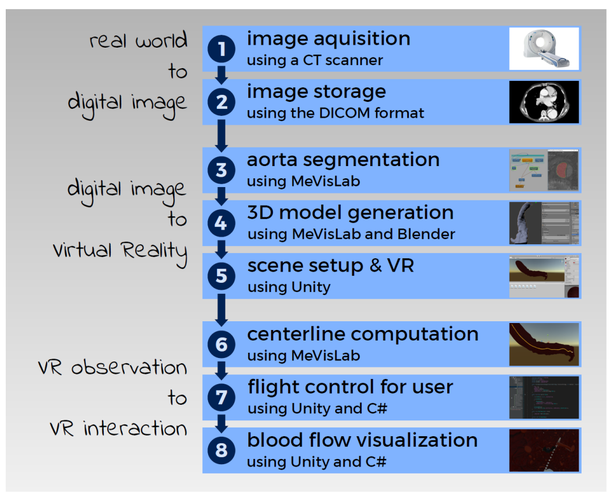
Methods

Visualizing a dissected aorta in VR comprises several steps. Figure 1 shows the basic workflow of the visualization process on the left side. It starts with the image acquisition using a CT scanner in step 1 and ends with the blood flow visualization using Unity in step 8. As shown in the Figure 1, different tools and application have been used for the intermediate steps 2-7, e.g. MeVisLab, for a segmentation and the 3D model generation.

A flight control is relevant for both, the player and the blood cells. However, even though player and blood cells have a similar behavior, there are some differences. The major difference is that while blood cells fly automatically, a player flight is supposed to be controlled by the user. Therefore, a framework has been implemented to keep the flight control as abstract and extensible as possible. Figure 1 shows the basic architecture of the flight control framework on the right side.

The path (*Path*) is a simple class wrapping the aorta centerline, a Unity *GameObject* consisting of multiple spheres. It extends the *GameObject* by the ability to hop back and forth between adjacent nodes (*next()*, *previous()*) and implements some more functions to provide information about the current node and whether it is the start node or the end node of the path.

The *Player* is the object that actually moves on a path. It is used by the aforementioned *FlightController*. It can perform operations such as moving (*move()*), increasing the speed of flight (*faster()*), decreasing the speed of flight (*slower()*) or flipping its flight direction (*flip()*). A *Player* is just one out of many possible objects that could move along a path. There might be other objects, such as the blood cells. Therefore, an abstract class for traversing along a path is implemented (*PathTraverser*). The *Player* is a subclassed *PathTraverser*.



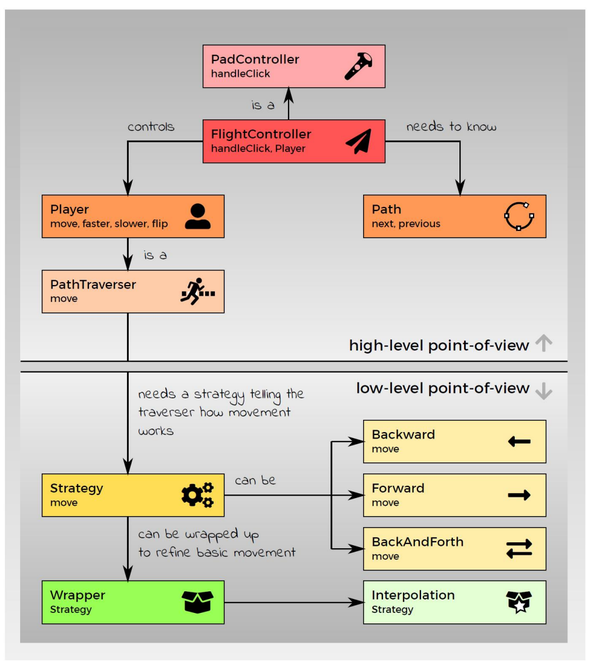


Figure 1 – Workflow of visualizing the aortic dissection in interactive Virtual Reality (**left**) and the flight control framework (**right**).

Results

VR applications require hardware with high performance. In our case the HTC Vive has been used for implementation and testing our work. It consists of a head mounted display (HMD), two wireless handheld controllers and two base stations called Lighthouses.

Given this environment, framerates have been measured using Unity's built-in profiler and stats-panel. For measuring frametime, dropped frames and warp misses, FCAT VR Capture Version 3.26.0.0, a tool provided by NVIDIA, has been used. FCAT VR Capture comes with a tool called FCAT VR Analyzer. It has been used to visualize the results yielded using FCAT VR Capture. The Analyzer is based on Python and requires PyQtGraph to be installed. For this work Anaconda 3.6.5 shipped with Python 3.6.3 has been used. The installed version of PyQtGraph has been 0.10.0.

Evaluation has been performed using different number of blood cells measuring overall memory consumption, average framerate, average frame timings, dropped frames, and warp misses for 60 seconds. Interval plots for a setting with zero blood cells, showed that all frames have successfully been generated while interval plots for 5,000 blood cells showed a high number of dropped frames. Plots showed, that frametimes drop after initialization phase. This effect may be attributed to caching. In addition, the frametimes are oscillating at high frequency. Analyzing the running scene using Unity revealed that the oscillation is caused by the garbage collector, which runs at scheduled time steps. In summary, a good performance providing a smooth user experience has been achieved with up to 2,200 blood cells while 5,000 blood cells had been far too much for rendering the scene smoothly (Figure 2).

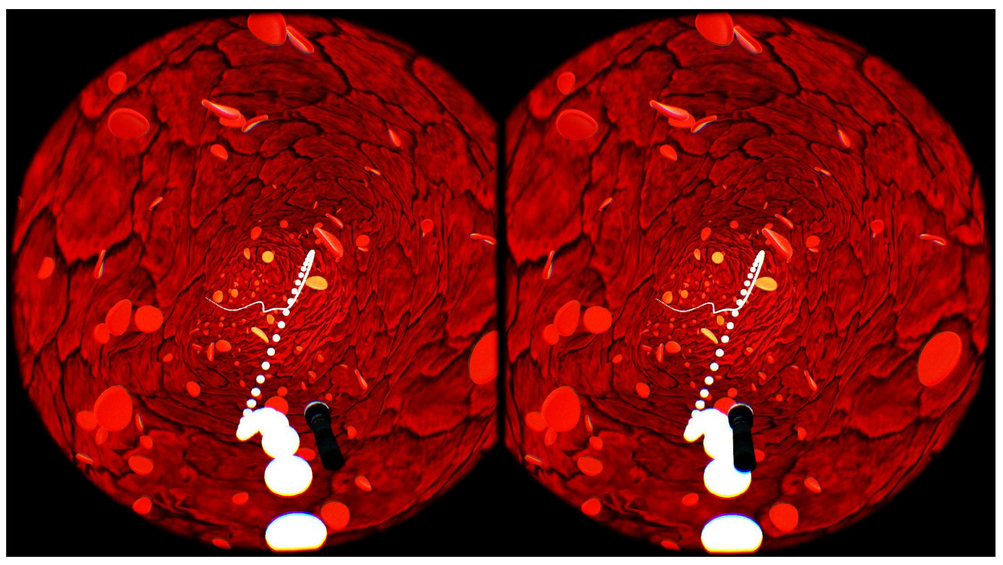


Figure 2 – VR-based simulation and visualization: The user can move for- and backward along the centerline of the aorta (white). Please see also the corresponding video on YouTube: https://www.youtube.com/c/JanEgger/videos

Conclusions

Summing up, the current solution introduced a workflow and framework for a VR visualization of AD. However, to develop a visualization that can be used in daily clinical routine, a systematic in-depth user evaluation has to be performed. In the end, a realistic simulation and visualization of AD could be a crucial step in getting a deeper insight and understanding of this manifold and complex disease that is still highly lethal. Finally, we want to point the interested reader to our cloud-based medical platform Studierfenster (www.studierfenster.at), which also enables to view medical data in Virtual Reality, e.g. the Google Cardboard.

Acknowledgements

TU Graz Lead Project "*Mechanics, Modeling and Simulation of Aortic Dissection*", Austrian Marshall Plan Foundation Scholarship 942121022222019, KLI 678-B31 (FWF), CAMed (COMET K-Project 871132), National Natural Science Foundation of China (81971709; 81828003; M-0019; 8191101624), Foundation of Ministry of Education of China Science and Technology Development Center (2018C01038), Foundation of Science and Technology Commission of Shanghai Municipality (19510712200), Shanghai Jiao Tong University Foundation on Medical and Technological Joint Science Research (ZH20182DA15, YG2019ZDA06, ZH2018QNA23) and the Overseas Visiting Scholars Program from the Shanghai Jiao Tong University (SJTU) in China.

References

1. Renapurkar, R. D. et al. "Aortic volume as an indicator of disease progression in patients with untreated infrarenal abdominal aneurysm," European Journal of Radiology 81 (2012) e87–e93 (2012).
2. Erbel, R. et al. "Diagnosis and management of aortic dissection," European Society of Cardiology. European Heart Journal, 22(18):1642-1681 (2001).
3. Egger, J. et al., "HTC Vive MeVisLab integration via OpenVR for medical applications," PLoS ONE 12(3): e0173972 (2017).
4. Egger, J. et al., "Integration of the OpenIGTLink Network Protocol for Image-Guided Therapy with the Medical Platform MeVisLab," Int J Med Robot. 8(3):282-90 (2012).
5. Pepe, A. et al. "IRIS: Interactive Real-Time Feedback Image Segmentation with Deep Learning," SPIE Medical Imaging, Biomedical Applications in Molecular, Structural, and Functional Imaging (2020).
6. Li, J. et al. "Towards Automatic Measurement of Type B Aortic Dissection Parameters: Methods, Applications and Perspective," Intravascular Imaging and Computer Assisted Stenting and Large-Scale Annotation of Biomedical Data and Expert Label Synthesis, Springer, pp. 64-72 (2018).

SPIE. MEDICAL IMAGING

Marriott Marquis Houston
Houston, Texas, United States
15 - 20 February 2020