

# Integration of the HTC Vive into the Medical Platform MeVisLab

Jan Egger<sup>a,b</sup>, Markus Gall<sup>a</sup>, Jürgen Wallner<sup>c</sup>, Pedro de Almeida Germano Boechat<sup>a</sup>  
Alexander Hann<sup>d</sup>, Xing Li<sup>c</sup>, Xiaojun Chen<sup>e</sup>, Dieter Schmalstieg<sup>a</sup>

<sup>a</sup> TU Graz, Institute for Computer Graphics and Vision, Inffeldgasse 16c/II, 8010 Graz, Austria

<sup>b</sup> BioTechMed-Graz, Krenngasse 37/1, 8010 Graz, Austria

<sup>c</sup> Medical University of Graz, Department of Maxillofacial Surgery, Auenbruggerplatz 12, 8036 Graz, Austria

<sup>d</sup> Department of Internal Medicine I, Ulm University, Albert-Einstein-Allee 23, 89081 Ulm, Germany

<sup>e</sup> Shanghai Jiao Tong University, School of Mechanical Engineering, 800 Dong Chuan Road, Shanghai 200240, China

## ABSTRACT

Virtual Reality (VR) is an immersive technology that replicates an environment via computer-simulated reality. VR gets a lot of attention in computer games but has also great potential in other areas, like the medical domain. Examples are planning, simulations and training of medical interventions, like for facial surgeries where an aesthetic outcome is important. However, importing medical data into VR devices is not trivial, especially when a direct connection and visualization from your own application is needed. Furthermore, most researchers don't build their medical applications from scratch, rather they use platforms, like MeVisLab, Slicer or MITK. The platforms have in common that they integrate and build upon on libraries like ITK and VTK, further providing a more convenient graphical interface to them for the user. In this contribution, we demonstrate the usage of a VR device for medical data under MeVisLab. Therefore, we integrated the OpenVR library into MeVisLab as an own module. This enables the direct and uncomplicated usage of head mounted displays, like the HTC Vive under MeVisLab. Summarized, medical data from other MeVisLab modules can directly be connected per drag-and-drop to our VR module and will be rendered inside the HTC Vive for an immersive inspection.

**Keywords:** Virtual Reality (VR), HTC Vive, Integration, MeVisLab, Medical Applications.

## 1. DESCRIPTION OF PURPOSE

In contrast to Augmented Reality (AR) [1], which aims to present information that is directly registered to the physical environment, Virtual Reality (VR) places a user inside a computer-generated environment. However, VR became recently increasingly popular, due to the fact that computer graphics have progressed to a point where the images are often indistinguishable from the real world. Thereby, the computer-generated images presented in games, movies, and other media are detached from our physical surroundings and led to new consumer devices such as head-mounted displays (HMD) and gesture-tracking devices, like the Oculus Rift or HTC Vive. Thus, VR not only places a user inside a computer-generated environment, rather it can completely immerse a user in a virtual world, as it removes any restrictions as to what a user can do or experience in VR [2]-[4]. However, beside games, movies, and other media, the medical field has great potential for the newly released HMDs, because they are now able to process and display high resolution medical data and applications acquired with modern Computed Tomography (CT) [5]-[8] and Magnetic Resonance Imaging (MRI) [9]-[25] scanners. This means these new technologies can have a major impact on health care in the near future [26]. Examples are preoperative surgery planning [27] and surgery simulators and trainers [28]. Nevertheless, others working in the field of medical VR are Nunnerley et al. [29] who tested the feasibility of an immersive 3D virtual reality wheelchair training tool for people with spinal cord injury (SCI). Therefore, they designed a Wheelchair Training System that used the Oculus Rift headset and a Dynamic Control wheelchair joystick. Newbutt et al. [30] studied the usage of a Virtual Reality Headset (Oculus Rift) in autism populations. In summary, the study explored the willingness, acceptance, sense of presence and immersion of ASD (autism spectrum disorder) participants. In [31] Farahani et al. explored the virtual reality technology and the Oculus Rift for the examination of digital pathology slides. They used an Oculus Rift Development Kit 2 (DK2) and a virtual desktop software, so three pathologists could review lymph node cases for digital pathology. However, to the best of our knowledge, there is no work that has integrated the HTC Vive into MeVisLab (<http://www.mevislab.de/>) [32]. Thus we introduce the HTC Vive to the medical domain, in addition and complementary to previous studies that worked with the Oculus Rift. In doing so, we developed a new module for the medical prototyping platform MeVisLab that provides an interface via the OpenVR library (<https://github.com/ValveSoftware/openvr>) to head mounted devices, therefore, enabling already the direct and uncomplicated usage of the HTC Vive under MeVisLab. Finally, and in contrast to the current version of the Oculus Rift, the HTC Vive uses a whole room to turn it into 3D space, thus enabling to walk around an object, which enables a deeper inspection.

## 2. METHODS

Data – As datasets for testing and evaluating the integration we used several high resolution Computed Tomography (CT) acquisitions from the clinical routine. The datasets consisted of 512x512 voxels in x- and y-direction with a few hundred slices in z-direction. We used datasets with variations in anatomy and location of the pathology, like skulls from patients, suffering from cranial defects. These datasets are also freely available for download and usage in own research purposes, but we kindly ask to cite our work if you do so [33], [34]:

[https://www.researchgate.net/publication/303371734\\_Cranial\\_Defect\\_Datasets](https://www.researchgate.net/publication/303371734_Cranial_Defect_Datasets)

Note: The datasets we received from our clinical partners have not been downsampled or altered in any way for testing the visualization of the original sized scans inside the HTC Vive. Thus, assessing the visual VR quality and evaluate the frames per second (fps) performance.

Workflow – A high level workflow diagram showing the communication and interaction between MeVisLab and the HTC Vive via OpenVR is presented in Figure 1. Thereby, MeVisLab provides several modules to import/load (medical) data, e.g. in the DICOM format. Afterwards, these image data can be processed and visualized with a variety of modules, depending on the task. That can be the segmentation of certain anatomical or pathological structures and the representation of the results in 2D and 3D viewer, or the processing with MeVisLab modules that base directly on ITK and VTK functions. In this study, the HTC Vive module communicates and interacts via OpenVR with the VR device, which is in the current implementation the HTC Vive. However, because the OpenVR API provides a way to connect and interact with Virtual Reality displays without relying on a specific hardware vendor's SDK, the module could also communicate with another VR device, like the Oculus Rift. In any case, the HTC Vive module sends the image data to the VR device to be rendered inside the HMD and the VR device provides the position and the orientation of the HMD to the MeVisLab module. This allows further visualization under MeVisLab, like rendering the view of the user wearing the HMD in a standard 3D viewer for other users not wearing a HMD. For the implementation on the VR side we followed, adapted and enhanced the OpenVR example `hellovr_opengl`. Thereby having an initialization method to load the SteamVR Runtime and render a distortion view of the left and right eye.

Network – The overall MeVisLab network with our HTC Vive module is presented in Figure 2. In this network, the medical data is loaded via a WEMLoad module (named DataLoad) and is directly passed to the HTC Vive module (rectangle input at the bottom of the HTC Vive module). Furthermore, the data is passed via a WEMModify, a SoWEMRenderer and a SoSeparator module to another SoSeparator (named 3DUserView) on the right side of the network. The window on the right side belongs to the 3DUserView module and shows what the user sees who wears the HTC Vive. Therefore, the HTC Vive module provides the HMD tracking coordinates of the HTC Vive to have the same position and viewing direction as the user wearing the HTC Vive (the HMD tracking coordinates are transferred via a parameter connection). The other modules are mainly arithmetic and matrices modules to decompose and compose the rotation and translation for a correct visualization. The HTC Vive module interface and its parameters are shown on the left side, further, a status window of the SteamVR is shown on the lower right corner, indicating that the HMD, the two controllers and the two Lighthouse base stations are ready (green).

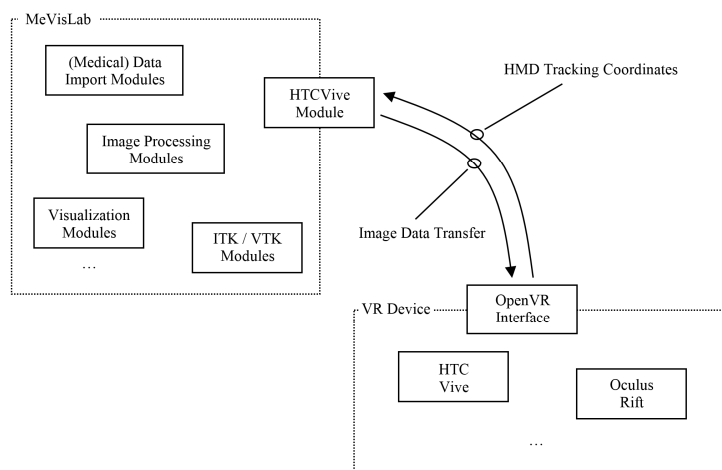


Fig. 1: High level workflow diagram showing the communication/interaction between MeVisLab and the HTC Vive via OpenVR.

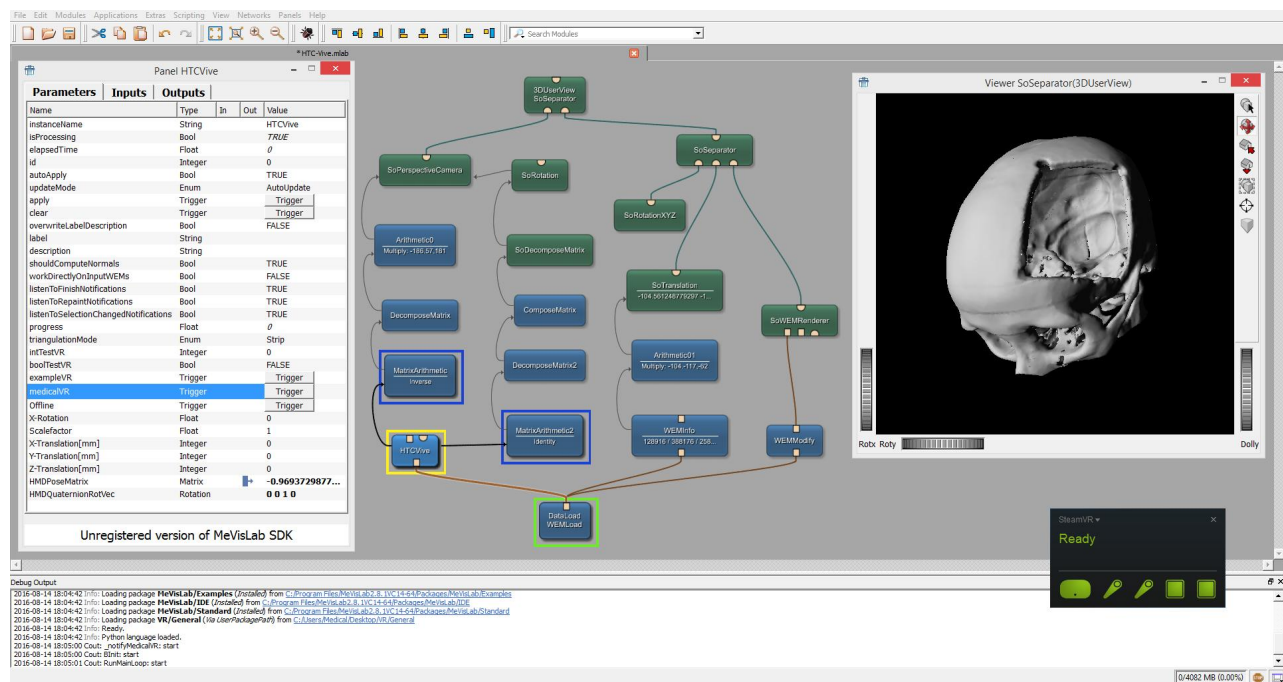


Fig. 2: The overall MeVisLab network with the HTC Vive module and its interface and parameters on the left side. In addition, our module provides the HMD tracking coordinates of the HTC Vive that can be used to have the same position and viewing direction (as the user wearing the HTC Vive) in a 3D view window on the right side.

### 3. RESULTS

Overall, the goal of this contribution was to investigate the feasibility of using the HTC Vive under the medical prototyping platform MeVisLab. Thus enabling the direct visualization of medical data in the head mounted device via the OpenVR library. The integration could be successfully achieved under Microsoft Windows 8.1 with the MeVisLab 2.8.1 (21-06-2016) Version for Windows Visual Studio 2015 X64 (<http://www.mevislab.de/download/>) and OpenVR SDK 1.0.2 (<https://github.com/ValveSoftware/openvr>). As MeVisLab module we implemented an image processing (ML) module in C++ using the Microsoft Visual Studio 2015 Community Edition and the MeVisLab Project Wizard. The HTC Vive MeVisLab module had an input for the medical data that was transferred and displayed in the HTC Vive. Further we provided the data at an output for visualization in a standard undistorted 3D view. In addition, our module provides the HMD tracking coordinates of the HTC Vive that can be used to have the same position and viewing direction (as the user wearing the HTC Vive) also in a 3D view (Figure 2, right window). Furthermore, an extra window can be created for a distorted view of the VR input for the right and left eye. For evaluation we tested several medical datasets, like patient skulls with cranial defects. As hardware setting for the evaluation we used a desktop PC with Windows 8.1 Enterprise installed. The hardware configuration of the desktop computer consisted of an Intel Core i7-3770 CPU @ 3.40GHz, 16 GB RAM and a NVIDIA GeForce GTX 970 graphics card. Figure 3 shows the frame timing results of the configuration for the CPU and the GPU. The frame timing evaluating comes directly from the SteamVR and can be activated and displayed via the SteamVR status window (Settings/Performance). The default frame timing view splits out CPU and GPU performance in a pair of stacked graphs. Thereby, the blue sections are the time spent by the application and further split between scene and other. *Scene* is the amount of work performed between when *WaitGetPoses* returns, and the second eye texture is submitted and *other* is any time spent after this for rendering the application's companion window, etc. The CPU timing here does not capture any parallel work being performed, for example on your application's main thread. The *Other* (brown) section in the GPU timing reflects any GPU bubbles, context switching overhead, or GPU work from other application getting scheduled in between other segments of work for that frame. *Late Starts* (indicated in red), is a special case where the application has exceeded its CPU budget for the frame, and is likely to drop a frame. However, according to the description on Valve's website, our system runs with 90 frames per second, since no frame is dropped in this diagram (note: an even more detailed examination can be achieved using *gpview*). For more information, please see (last access December 2016):

[https://developer.valvesoftware.com/wiki/SteamVR/Frame\\_Timing](https://developer.valvesoftware.com/wiki/SteamVR/Frame_Timing)

As seen in the frame timing diagram, we could achieve a sufficient framerate for the desktop PC configuration, where 10 frames per second are already considered as real-time or interactive in computer graphics applications [35]. Subsequently, the medical dataset could be viewed very pleasant to the human eye inside the HTC Vive.

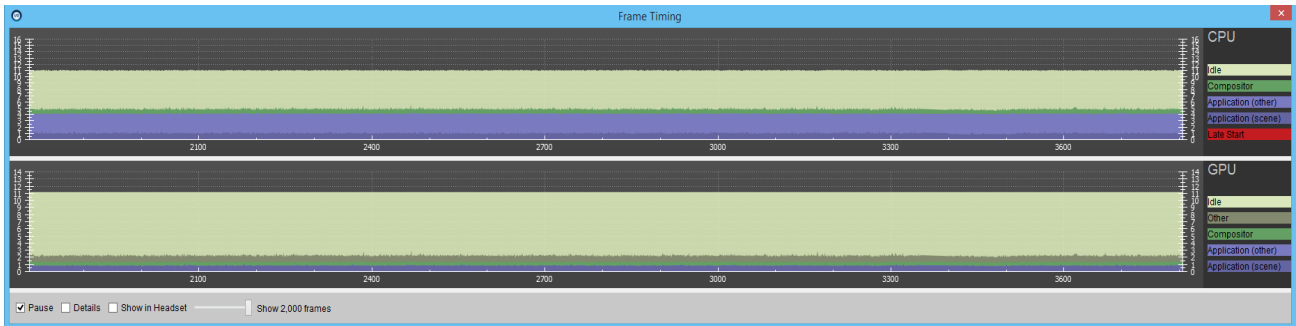


Fig. 3: Frame timing results of the CPU and GPU for a desktop PC with Intel Core i7-3770 CPU @ 3.40GHz, 16 GB RAM and a NVIDIA GeForce GTX 970 graphic card.

Example 2: Intestinum

Name	Type	In	Out	Value
listenToFinishNotifications	Bool			TRUE
listenToRepairNotifications	Bool			TRUE
listenToSelectionChangedNotifications	Bool			TRUE
progress	Float			0
triangulationMode	Enum			Strip
intTestVR	Integer			0
boolTestVR	Bool			FALSE
exampleVR	Trigger			Trigger
medicalVR	Trigger			Trigger
Offline	Trigger			Trigger
X-Rotation	Float			0
Scalefactor	Float			1
X-Translation(mm)	Integer			0
Y-Translation(mm)	Integer			0
Z-Translation(mm)	Integer			0
HMDPoseMatrix	Matrix			-0.96937298...
HMDQuaternionRotVec	Rotation			0 0 1 0

Video 1: Demonstration of the integration of the HTC Vive into the medical platform MeVisLab  
<http://dx.doi.org/10.1117/12.2263234>

## 4. CONCLUSIONS

In this contribution, we presented the successful integration and evaluation of the HTC Vive into the medical prototyping platform MeVisLab via the OpenVR library. OpenVR is a software development kit and application programming interface developed by Valve for supporting the SteamVR (HTC Vive) and other virtual reality headset devices. Thus, the demonstrated integration is in principle device and vendor independent and can also be used with other devices. The integration had been carried out with the C++ implementation of a MeVisLab image processing (ML) module. Providing inputs for medical datasets, the module is easy to use and can be applied in new MeVisLab networks or added to existing ones. Connecting data outputs under MeVisLab per drag and drop to an input of our module is already enough to get the data into the HTC Vive. Dataset operations and manipulations, like translation, rotation, segmentations, etc. which can be performed in advance with the corresponding and existing MeVisLab modules will directly be shown also in VR. In summary, the achieved research highlights of the presented work are:

- The successful integration of OpenVR with MeVisLab has been demonstrated;
- The developed solution allows MeVisLab programs to connect to virtual reality headset devices;
- Real-time visualization of medical data in VR is now possible under MeVisLab;
- For proof of concept, the integration has been tested with the HTC Vive device;
- The HTC Vive module can be used in new MeVisLab networks or added to existing ones.

There are several areas for future work, in particular the evaluation of our integration with a greater amount of medical data formats and supporting these with our module, and providing it to the research community. Additionally, the development of a MeVisLab module communicating with the Oculus Rift. Furthermore, using the integration to support computer-aided reconstruction of facial defects [36], [37] with photorealistic rendering in VR [38], enabling an even more realistic assessment of the pre-operative planning results, which also applies in other medical image processing areas [39]-[48]. Moreover, performing image-guided therapy tasks in VR, like the planning of facial surgery using miniplates [49]. Finally, integrating OpenVR into other research platforms, like (3D) Slicer [50], [51] and the two Medical Imaging (Interaction) Toolkits (MITK) from Heidelberg in Germany ([www.mitk.org](http://www.mitk.org)) and Beijing in China ([www.mitk.net](http://www.mitk.net)), respectively.

## ACKNOWLEDGEMENT

The work received funding from BioTechMed-Graz in Austria (“*Hardware accelerated intelligent medical imaging*”), the 6<sup>th</sup> Call of the Initial Funding Program from the Research & Technology House (F&T-Haus) at the Graz University of Technology (PI: Jan Egger, Ph.D., Ph.D.) and ClinicIMPACT (610886). Dr. Xiaojun Chen receives support by the Natural Science Foundation of China (Grant No.: 81511130089) and the Foundation of Science and Technology Commission of Shanghai Municipality (Grants No.: 14441901002, 15510722200 and 16441908400). A video demonstrating the integration of the HTC Vive into the medical platform MeVisLab is available under the following YouTube channel:

<https://www.youtube.com/c/JanEgger/videos>

## REFERENCES

- [1] Chen, X. et al. “Development of a Surgical Navigation System based on Augmented Reality using an Optical see-through Head-mounted Display,” *J Biomed Inform.* 2015 Jun;55:124-31. doi: 10.1016/j.jbi.2015.04.003 (2015).
- [2] Schmalstieg, D. & Höllerer, T. “Augmented Reality: Principles and Practice,” Addison-Wesley Professional; 1<sup>st</sup> ed., Paperback, 528 pages, ISBN 978-0321883575 (2016).
- [3] Bowman, D. A. & McMahan, R. P. “Virtual reality: How much immersion is enough?,” *IEEE Computer* 40(7):36-43 (2007).
- [4] Rheingold, H. “Virtual Reality: Exploring the Brave New Technologies,” Simon & Schuster Adult Publishing Group, ISBN:0671693638 (1991).
- [5] Egger, J. et al “Interactive Volumetry of Liver Ablation Zones,” *Sci Rep* 5:15373 (2015).
- [6] Egger, J. et al. “Aorta Segmentation for Stent Simulation,” 12<sup>th</sup> International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI), Cardiovascular Interventional Imaging and Biophysical Modelling Workshop, 10 pages, London, UK (2009).
- [7] Egger, J. et al. “Modeling and Visualization Techniques for Virtual Stenting of Aneurysms and Stenoses,” *Computerized Medical Imaging and Graphics*, 36(3), pp. 183-203 (2012).
- [8] 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).
- [9] Zukic, D. et al. “Segmentation of Vertebral Bodies in MR Images,” *Vision, Modeling, and Visualization (VMV)*, The Eurographics Association, pp. 135-142, (2012).

- [10] Egger, J. et al. "Pituitary Adenoma Segmentation," In: Proceedings of International Biosignal Processing Conference, Charité, Berlin, Germany (2010).
- [11] Egger, J. et al. "A flexible semi-automatic approach for glioblastoma multiforme segmentation," Proceedings of International Biosignal Processing Conference, Charité, Berlin, Germany (2010).
- [12] Egger, J. et al. "A Comparison of Two Human Brain Tumor Segmentation Methods for MRI Data," Proceedings of 6th RB Conference on Bio-Medical Engineering, State Technical University, Moscow, Russia, 4 pages (2010).
- [13] Egger, J. et al. "A medical software system for volumetric analysis of cerebral pathologies in magnetic resonance imaging (MRI) data," *Journal of medical systems* 36 (4), 2097-2109 (2012).
- [14] Zukic, D. et al. "Robust Detection and Segmentation for Diagnosis of Vertebral Diseases using Routine MR Images," *Computer Graphics Forum*, Volume 33, Issue 6, Pages 190–204 (2014).
- [15] Egger, J. et al. "PCG-Cut: Graph Driven Segmentation of the Prostate Central Gland," *PLOS ONE* 8 (10), e76645 (2013).
- [16] Schwarzenberg, R. et al. "A Cube-Based Approach to Segment Vertebrae in MRI-Acquisitions," Proceedings of Bildverarbeitung für die Medizin (BVM), Springer Press, pp. 69-74 (2013).
- [17] Schwarzenberg, R. et al. "Cube-Cut: Vertebral Body Segmentation in MRI-Data through Cubic-Shaped Divergences," In: *PLoS One* (2014).
- [18] Egger, J. et al. "Template-Cut: A Pattern-Based Segmentation Paradigm," *Sci Rep* 2:420 (2012).
- [19] Egger, J. et al. "Interactive-cut: Real-time feedback segmentation for translational research," *Computerized Medical Imaging and Graphics* 38 (4), 285-295 (2014).
- [20] Zukić, D. et al. "Glioblastoma Multiforme Segmentation in MRI Data with a Balloon Inflation Approach," Proceedings of 6th RB Conference on Bio-Medical Engineering, State Technical University, Moscow, Russia, pp. 4 pages (2010).
- [21] Egger, J. "Refinement-Cut: User-Guided Segmentation Algorithm for Translational Science," *Sci Rep* 4:5164 (2014).
- [22] Egger, J. et al. "Square-cut: a segmentation algorithm on the basis of a rectangle shape," *PLoS One* 7, e31064; DOI:10.1371/journal.pone.0031064 (2012).
- [23] Egger, J. et al. "Nugget-Cut: A Segmentation Scheme for Spherically- and Elliptically-Shaped 3D Objects," 32nd Annual Symposium of the German Association for Pattern Recognition (DAGM), LNCS 6376, pp. 383–392, Springer Press, Darmstadt, Germany (2010).
- [24] Egger, J. et al. "Manual refinement system for graph-based segmentation results in the medical domain," *Journal of medical systems* 36 (5), 2829-2839 (2012).
- [25] Egger, J. et al. "Graph-Based Tracking Method for Aortic Thrombus Segmentation," Proceedings of 4<sup>th</sup> European Congress for Medical and Biomedical Engineering, Engineering for Health, Antwerp, Belgium, Springer, pp. 584-587 (2008).
- [26] McCloy, R. & Stone, R. "Virtual reality in surgery," *BMJ*. 323(7318):912-915 (2001).
- [27] Reitinger B, et al. "Liver surgery planning using virtual reality," *IEEE Comput Graph Appl*. 26(6):36-47 (2006).
- [28] Gor, M., McCloy, R., Stone, R. & Smith, A. "Virtual reality laparoscopic simulator for assessment in gynaecology," *BJOG*. 110(2):181-7 (2003).
- [29] Nunnerley, J., Gupta, S., Snell, D. & King, M. "Training wheelchair navigation in immersive virtual environments for patients with spinal cord injury - end-user input to design an effective system," *Disabil Rehabil Assist Technol*. 4:1-7 (2016).
- [30] Newbutt, N. et al. "Brief Report: A Pilot Study of the Use of a Virtual Reality Headset in Autism Populations," *J Autism Dev Disord*. (2016).
- [31] Farahani, N. et al. "Exploring virtual reality technology and the Oculus Rift for the examination of digital pathology slides," *J Pathol Inform*. 7:22. doi: 10.4103/2153-3539.181766. eCollection (2016).
- [32] Egger, J. et al. "Integration of the OpenIGTlink network protocol for image guided therapy with the medical platform MeVisLab," *The International Journal of medical Robotics and Computer assisted Surgery*, 8(3):282-390 (2012).
- [33] Gall, M., Li, X., Chen, X., Schmalstieg, D. & Egger, J. "Cranial Defect Datasets," *ResearchGate* (2016).
- [34] Gall, M., et al. "Computer-Aided Planning and Reconstruction of Cranial 3D Implants," *IEEE Engineering in Medicine and Biology Society (EMBC'16)*, Orlando, FL, August USA, IEEE Press, pp. 1179-1183 (2016).
- [35] Ai, Z., et al. "Cranial implant design using augmented reality immersive system," *Stud Health Tech Inform*. 125:7-12 (2007)
- [36] Gall, M. et al. "Computer-Aided Planning of Cranial 3D Implants," *Int J CARS* 11 (Sup. 1):S241, Heidelberg, Germany (2016)
- [37] Li, X. et al. "A semi-automatic implant design method for cranial defect restoration," *Int J CARS* 11 (Suppl. 1):S241-S243, Heidelberg, Germany (2016).
- [38] Vogelreiter, P. et al. "Global Illumination Rendering for High-Quality Volume Visualization in the Medical Domain," *Face 2 Face – Science meets Arts*, Medical University of Graz, Department of Maxillofacial Surgery (2015).
- [39] Kuhnt, D. et al. "Fiber tractography based on diffusion tensor imaging (DTI) compared with High Angular Resolution Diffusion Imaging (HARDI) with compressed sensing (CS) – initial experience and clinical impact," *Neurosurgery*, Volume 72, pp. A165-A175 (2013).
- [40] Egger, J., Großkopf, S. and Freisleben, B. "Simulation of bifurcated stent grafts to treat abdominal aortic aneurysms (AAA)," *Proc. SPIE* 6509, *Medical Imaging 2007: Visualization and Image-Guided Procedures*, 65091N (22 March 2007); doi: 10.1117/12.709260 (2007).
- [41] Egger, J. et al. "Fast self-collision detection and simulation of bifurcated stents to treat abdominal aortic aneurysms (AAA)," 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Lyon, France, pp. 6231-6234, IEEE Press (2007).

- [42] Lu, J. et al. "Detection and visualization of endoleaks in CT data for monitoring of thoracic and abdominal aortic aneurysm stents," Proceedings of SPIE Medical Imaging Conference, Vol. 6918, pp. 69181F(1-7), San Diego, USA, (2008).
- [43] Bauer, M. et al. "A fast and robust graph-based approach for boundary estimation of fiber bundles relying on fractional anisotropy maps," 20th International Conference on Pattern Recognition (ICPR), Istanbul, Turkey, pp. 4016-4019 (2010).
- [44] Bauer, M. et al. "Boundary estimation of fiber bundles derived from diffusion tensor images," International journal of computer assisted radiology and surgery 6 (1), 1-11 (2011).
- [45] Greiner, K. et al. "Segmentation of Aortic Aneurysms in CTA Images with the Statistic Approach of the Active Appearance Models," Proceedings of Bildverarbeitung für die Medizin (BVM), Berlin, Germany, Springer Press, pages 51-55 (2008).
- [46] Egger, J. et al. "Preoperative Measurement of Aneurysms and Stenosis and Stent-Simulation for Endovascular Treatment," IEEE International Symposium on Biomedical Imaging: From Nano to Macro, Washington (D.C.), USA, pp. 392-395, IEEE Press (2007).
- [47] Egger, J., Mostfarkic, Z., Grosskopf, S. and Freisleben, B. "A Fast Vessel Centerline Extraction Algorithm for Catheter Simulation," 20th IEEE International Symposium on Computer-Based Medical Systems, Maribor, Slovenia, pp. 177-182, IEEE Press (2007).
- [48] Klein, J. et al. "On the Reliability of Diffusion Neuroimaging," Neuroimaging, INTECH, pp. 1-24, ISBN: 978-953-307-127-5 (2010).
- [49] Gall, M. et al. "Computer-aided Reconstruction of Facial Defects," IEEE Engineering in Medicine and Biology Society (EMBC'16), Posters, Orlando, FL, USA (2016).
- [50] Egger, J. et al. "GBM Volumetry using the 3D Slicer Medical Image Computing Platform," Sci Rep., Nature Publishing Group (NPG), 3:1364, 1-7 (2013).
- [51] Egger, J. et al. "Pituitary Adenoma Volumetry with 3D Slicer," PLOS ONE 7 (12), e51788 (2012).