Gestural 3D Interaction with a Beating Heart: Simulation, Visualization and Interaction

F. Ioakemidou[2], F. Ericson[2], J. Spühler[1], A. Olwal[3], J. Forsslund[2], J. Jansson[1], E.-L. Sallnäs Pysander[2] and J. Hoffman[1]

[1]Numerical Analysis, School of Computer Science and Communication, KTH, Sweden
[2] Human-Computer Interaction, School of Computer Science and Communication, KTH, Sweden
[3] Media Laboratory, School of Architecture and Planning, MIT, USA

Abstract

The KTH School of Computer Science and Communication (CSC) established a strategic platform in Simulation-Visualization-Interaction (SimVisInt) in 2009, focused on the high potential in bringing together CSC core competences in simulation technology, visualization and interaction. The main part of the platform takes the form a set of new trans-disciplinary projects across established CSC research groups, within the theme of Computational Human Modeling and Visualization: (i) interactive virtual biomedicine (HEART), (ii) simulation of human motion (MOTION), and (iii) virtual prototyping of human hand prostheses (HAND). In this paper, we present recent results from the HEART project that focused on gestural and haptic interaction with a heart simulation.

Keywords: *Virtual and Interactive Environments*

1. Heart

Cardiac disease is the major cause of death in western society. The vision of the HEART project is to build a virtual human heart from medical imaging data, together with an interactive interface (see Figure 1) with visual, haptic and sonic feedback. External collaborators include Umeå University, Linköping University, KTH School of Technology and Health, and the Barcelona Supercomputing Center. The basic idea is to use medical imaging technology to create a virtual model of the heart wall motion, blood flow dynamics, and mechanical stresses. Visual, haptic and sonic interaction with the heart model can be used to get an enhanced understanding of the heart function, and for help in diagnosis as a complement to medical imaging information. In addition, the interactive interface opens for modifications of the virtual heart, simulating cardiac disease or available treatment options, such as choosing between different mechanical heart valves, and thus offering a tool for decision support in planning surgery or other treatment.

Today the model takes the form of a left ventricle blood flow simulation, based on a geometrical model from patient-specific ultrasound data (Figure 2), including a visual and haptic interface [Aec09], [HJ06].

Figure 1: *3D interaction with a simulation of the blood flow in the left ventricle of the heart.*

2. The numerical simulation

The simulation is produced by first defining a moving geometry from measurements for which a mesh can be generated (described in subsection 2.1), the incompressible Navier-Stokes equations are then posed on the geometry (described in subsection 2.2), and numerically solved by a finite ele-

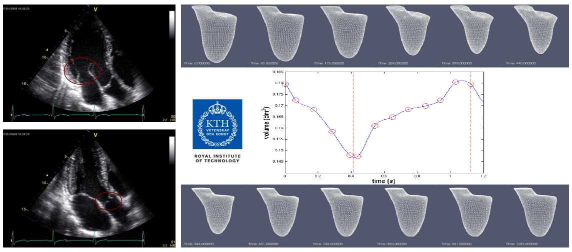

Figure 2: *Geometric model of the left ventricle constructed from ultrasound in collaboration with Umeå University (Mats G. Larson, Anders Waldenström, Ulf Gustafsson and Per Vesterlund).*

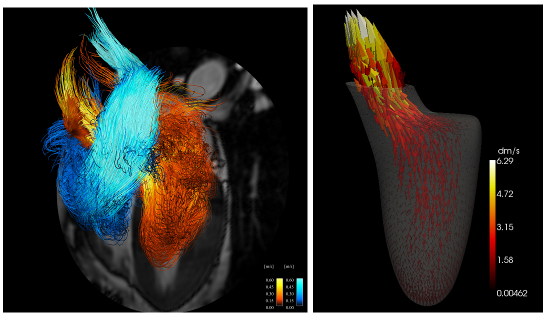

Figure 3: *MRI measurement (left) from Linköping University (Tino Ebbers), and computer simulation (right) of the blood flow in the heart in collaboration with KTH-STH (Lars-Åke Brodin and Michael Broomé).*

ment method (FEM) (described in subsection 2.3). The software implementation of the method is briefly described in subsection 2.4 together with the results.

2.1. From patient-specific ultrasound to the geometrical heart model

In our approach [Aec09] the blood in the left ventricle (LV) is driven by the prescribed movement of the inner heart wall. The model geometry of the LV is based on ultrasound measurements of the position of the inner wall at three different levels at twelve specific time points during the cardiac cycle. This is of relevance, as the movement of the heart wall is not uniform during the heart cycle that lasts about one second. Surface meshes of the chamber are constructed from these measured points, for each sample time, as shown in Figure 2. We then build a three-dimensional mesh of tetrahedrons at the initial time and apply Hermite interpolation to allocate the position of all boundary nodes at every time step. The Laplace mesh smoothing algorithm is used to deform the mesh to fit the given surface meshes during the cardiac cycle.

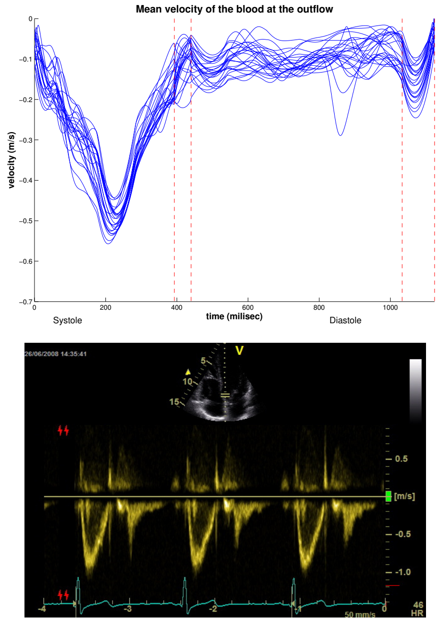

Figure 4: *Mean velocity measured at arbitrary points at the outflow in the simulation and velocity at the outflow derived with help of ultrasound measurements*

2.2. Mathematical model

We simulate the blood flow with the incompressible Navier-Stokes equations. The Arbitrary Lagrangian-Eulerian description [DHPRF04] is used to account for the movement of the mesh. We seek the velocity u and pressure p:

$$\begin{aligned} \dot{u} + ((u-\hat{w})\cdot\nabla)u - \nu\Delta u + \nabla p &= 0 \\ \nabla\cdot u &= 0 \end{aligned} \qquad (1)$$

with ν the kinematic viscosity and $\hat{w}$ the mesh velocity.

By defining different boundary conditions, we divide the cardiac cycle into the basic stages of diastole and systole. A no-slip boundary condition on the wall and closed valves is applied and the pressure is prescribed to model the inflow through the mitral valve and the outflow through the aortic valve.

2.3. FEM with Streamline Diffusion Stabilization

We apply the General Galerkin (G2) cG(1)cG(1) Finite Element Method (FEM) [HJ06] with piecewise continuous linear solution in time and space for solving the governing equation 1. For the standard FEM formulation of the model we only have stability of the discrete solution U but not of its spatial derivatives. This means that the solution can be oscillatory, causing inefficiency by introducing unnecessary error. We therefore choose a weighted standard

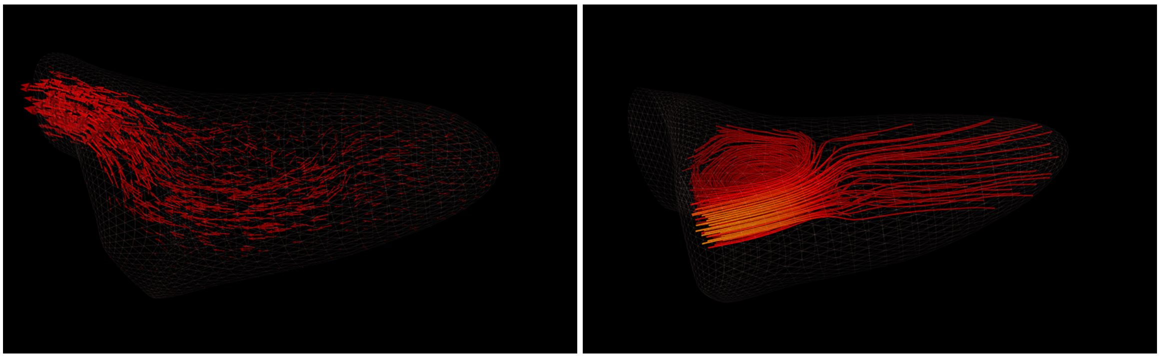

Figure 5: *Simulated blood flow velocity represented as directional 3D glyphs (left) and streamlines (right).*

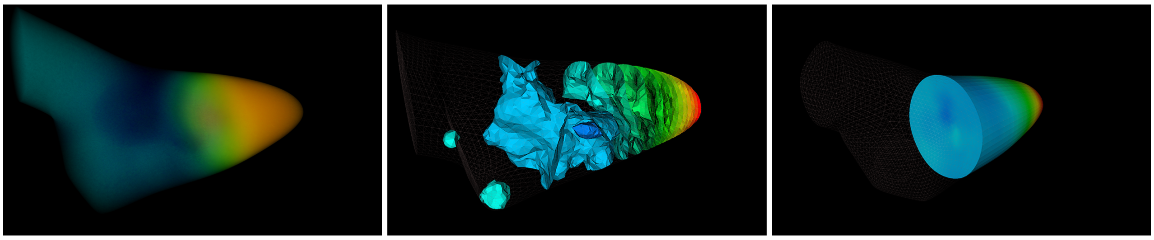

Figure 6: *The pressure data visualized with volume rendering (left), ISO surfaces (center), and using a cut plane for slicing (right).*

FEM/streamline diffusion method as follows: find the discrete velocity U and pressure P in the corresponding discrete function spaces V_0 and Q with test functions v and q such that

$$\begin{aligned} &((U^n - U^{n-1})k_n^{-1} + (\bar{U}^n - \hat{W}^{n-1})\cdot\nabla\bar{U}^n, v) \\ &+(2\nu\varepsilon(\bar{U}^n), \varepsilon(v)) - (P^n, \nabla\cdot v) + (\nabla\cdot\bar{U}^n, q) \\ &+\mathrm{SD}_\delta(\bar{U}^n, P^n; v, q) = 0 \quad \forall (v,q) \in V_0^n \times Q^n \end{aligned} \tag{2}$$

where $\bar{U}^n = 1/2\left(U^n + U^{n-1}\right)$, $\varepsilon(u) = \frac{1}{2}(\nabla u + \nabla u^T)$, k_n is the time step on time interval n, and with the stabilization term defined as

$$\begin{aligned} &\mathrm{SD}_\delta(\bar{U}^n, P^n; v, q) = \\ &(\delta_1((\bar{U}^n - \hat{W}^{n-1})\cdot\nabla\bar{U}^n + \nabla P^n - f), (\bar{U}^n - \hat{W}^{n-1})\cdot\nabla v + \nabla q) \\ &+ (\delta_2 \nabla\cdot\bar{U}^n, \nabla\cdot v) \end{aligned} \tag{3}$$

δ_1 and δ_2 are the stabilization parameters.

2.4. Simulation results

The simulations are done in DOLFIN [LW10], a differential equation problem solving environment, and UNICORN [HJNJ11], a unified continuum mechanics solver, which are developed as a part of the software project FEniCS and where we are active contributors.

The mean velocity at the outflow and inflow, as well as the pressure, are quantities that are interesting for medical examination and that can also be used for evaluation of the model. An example is shown in Figure 4.

3. Gestural interaction with a 3D Visualization

The interactive HEART demonstration is a public showcase designed to aid the perception and understanding of the simulation. Blood flow and pressure are visualized in 3D and can be manipulated using physical movement and gestures.

3.1. Visualization of blood flow

The system was developed using the Visualization Toolkit (VTK, http://www.vtk.org) which provides efficient implementations of many advanced graphics techniques. Our visualization animates deformations of the heart geometry throughout a heartbeat while illustrating other properties of the simulation.

The blood flow velocity field produced by solving eq. 2 in UNICORN is illustrated by arrows proportional in size to the flow velocity. Optionally flow can be visualized by animated streamlines, both alternatives shown in Figure 5.

The pressure data (also from eq. 2) is visualized using hardware accelerated volume rendering techniques. Optionally ISO surfaces can be used or an interactive cut plane. Alternatives shown in Figure 6.

The system also supports stereoscopic rendering with head tracking, as shown in Figure 7.

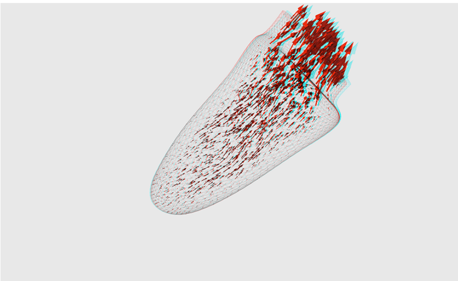

Figure 7: *The system also supports stereoscopic rendering, here enabled in anaglyph (red/cyan) mode.*

3.2. Gestural interaction using a depth-sensing camera

A depth-sensing camera (Microsoft Kinect) was used to acquire input for natural interaction. The OpenNI tracking library (http://www.openni.org) was combined with the OpenCV computer vision library (http://opencv.willowgarage.com) to recognize gestures and track the joints of users in 3D.

The OpenNI library provides real time 3D tracking of human body joints, of particular interest to us are hands and head. In order to provide the algorithm with necessary body metrics the user must first take the calibration pose shown in Figure 8.

Figure 8: *For skeleton tracking of the user, a one-time calibration pose is required initially.*

3.3. Detection of open/closed hands

The 3D tracking provides accurate positional data, which we use to manipulate parameters of the visualization. Since we track the user's hands constantly we need to provide a way for users to easily initiate / abort interaction with the simulation.

By swiping one closed hand in the air the user can turn the simulation around to inspect it from different viewpoints. The camera can also be zoomed out and in by closing both hands and moving them closer together or further apart respectively, as shown in Figure 9 and Figure 10.

Figure 9: *The user "grasps the heart" to rotate it with the right hand.*

The detection of open/closed hands is achieved by the combination of different techniques which aim to cover the majority of cases. First the a polygon approximation of the hand contour is acquired from the depth camera image, in which we can locate and isolate the hands using the information of the tracked joints. Similarity of the the hand contour area to the area of its convex hull is often a strong indication of a closed hand. To improve this estimate we analyze in detail the differences between the contour and hull. Finally we apply a custom algorithm to look for sharp consecutive changes of direction along the contour, usually signifying outstretched fingers. By experimenting with the parameters of these algorithms we arrived at a solution that makes accurate predictions in real time for most cases and does not produce any significant lag. See Figure 11.

3.4. Haptic feedback

Additional experiments were conducted using a desktop application built in C++ and CHAI 3D (http://www.chai3d.org/), which allowed the use of haptic devices (e.g., SensAble Phantom Omni, shown in Figure 12, left) for experiencing the dynamics in a heart beat. It allowed the user to move a 3D cursor inside the blood flow simulation and "feel" the movement of particles through the haptic feedback propagated through the device's stylus. The particular haptic device used could, however, only generate forces to control a single point's 3D position, which would be mapped to the stylus's tip. Haptic devices with 6 degrees of freedom (i.e., 3D position and 3D torque) are significantly more costly and complex, but most still only allow interaction at a single point.

3.5. Lo-fi haptics with motorized faders

As an alternative solution, a series of experiments were also conducted using a low-cost control fader device (Behringer BCF2000, shown in Figure 12, right) that has eight motorized faders. This device allowed multi-dimensional input (from the user) and output (from the simulation). The fader device was interfaced using the MIDI protocol and made it

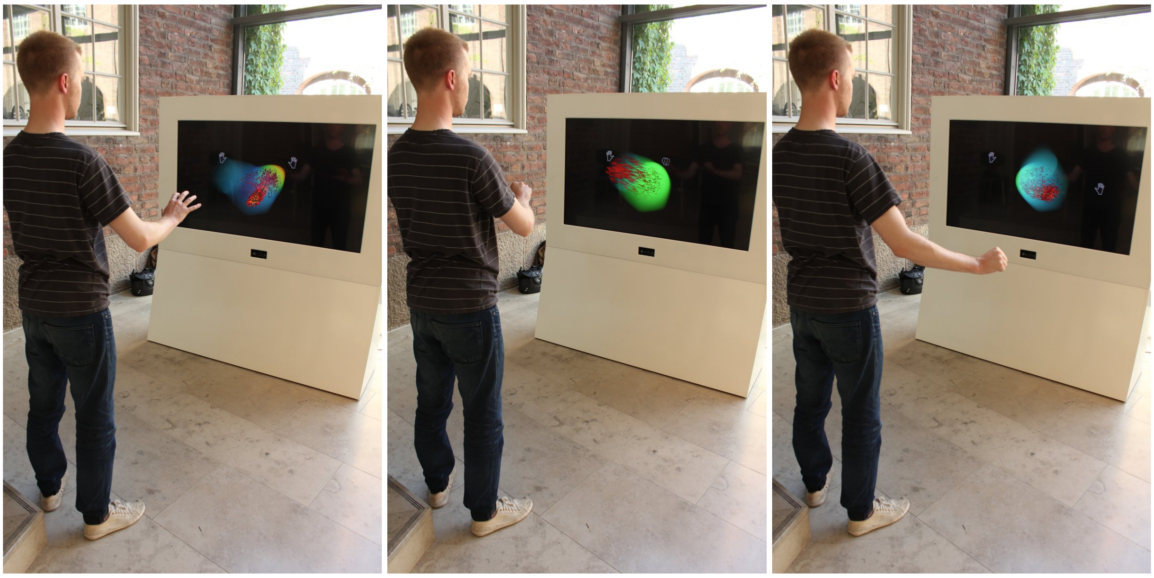

Figure 10: *From left to right: a) The user has both hands open, as also indicated by the on-screen hand symbols. b) The right hand is closed, which allows the user to rotate the model, by moving the closed hand. c) After moving the hand the user observes the simulation from another angle, the hand must now be opened to stop rotating.*

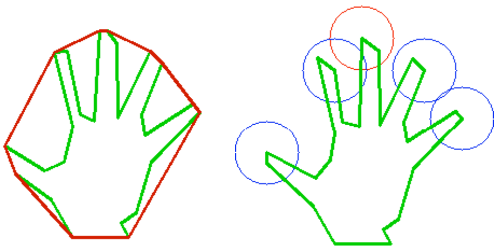

Figure 11: *Hands are analyzed by comparing the contour to its convex hull, and by using a finger detection algorithm. This makes it possible to establish whether hands are open or closed.*

possible to map various parameters to fader movement on the device. More motorized faders are straightforward to add by daisy-chaining multiple devices, which is natively supported by this type of the device as well as the MIDI protocol.

References

[Aec09] Aechtner M.: *Arbitrary Lagrangian-Eulerian Finite Element Modelling of the Human Heart*. Master's thesis, 2009. 1, 2

[DHPRF04] Donea J., Huerta A., Ponthot J.-P., Rodriguez-Ferran A.: *Arbitrary Lagrangian-Eulerian Methods*. Wiley, 2004, pp. 413–437. 2

[HJ06] Hoffman J., Johnson C.: *Computational Turbulent Incompressible Flow: Applied Mathematics Body and Soul Vol 4*. Springer-Verlag Publishing, 2006. 1, 2

[HJNJ11] Hoffman J., Jansson J., Nazarov M., Jansson N.: Unicorn: A Unified Continuum Mechanics Solver. In *Automated Scientific Computing*. Springer, 2011. 3

[LW10] Logg A., Wells G. N.: DOLFIN: Automated finite element computing. *ACM Trans. Math. Softw. 37*, 2 (2010), 1–28. 3

Figure 12: *Left) The Omni haptic device from SensAble uses a mechanically tracked stylus that provides 6 degrees of freedom input (3D position + 3D orientation) and force feedback in 3 dimensions. Right) The Behringer BCF2000 is a fader controller device, with eight motorized faders that can be mechanically controlled from a computer application using the MIDI protocol.*