MedVis Workshop - TU Delft 2014
A. Vilanova
(Supervisor)

Volume 1 (2014), Number 1

Web based blood-flow visualization using particles

David Hübner[1] and James Wafula[2]

[1]ID: 4320468, MSc Applied Mathematics (COSSE Yr 1), TU Delft
[2]ID: 4330889, MSc Applied Mathematics (COSSE Yr 1), TU Delft

Abstract

Medical studies have shown that Cardiovascular diseases (CVD) are the leading cause of death worldwide. The beginning and continued progression of CVD depends strongly on the blood flow patterns. The phenominal advances in technology in recent years have come along with the attendant result of growth in the availability of medical data. For example, 4D PC-MRI acquisition technologies have enabled reliable 3D flow measurements over time thereby allowing numerical and analytical analysis of patient-specific hemodynamics. There is active research in medical sciences on the relation between characteristic flow patterns and occurence and progression of diseases. The technique in use in practice has been the manual inspection of 2D slices which requires a full mental reconstruction of the unsteady blood-flow field, a tedious process requiring expert knowledge. In a previous work by Martijn Pieters, a WebGL based interactive blood-flow animation by using lines as visualization technique was implemented. In this work, we present an extension of Martijn Pieters web animation by implementing a particle-based visualization and test it on a patient and a volunteer dataset. We provide a discussion on why the particle model is appropriate and motivate the instances where a combination of particles and lines may be desirable. We also implement a fast and simple WebGL motion detector that uses the built-in webcam which gives us leverage to rotate the 3D mesh of the aorta without touching the mouse.

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

1. Introduction

Cardiovascular diseases (CVD) is a class of diseases that involve the heart and the blood vessels. CVD causes 47% of all deaths in Europe and 40% in the EU and is the leading cause of death worldwide [NTS*12]. The key to an appropriate treatment of the CVD is a precise diagnosis.

Traditional diagnosis of CVD is based on the medical and family history of the patient, different risk factors, physical examination and coordination and an inspection of the anatomical structure and function, whereas the blood-flow behavior is seldom analyzed. However, modern Magnetic Resonance Imaging (MRI) techniques enable the acquisition of high-quality data which can significantly contribute to the diagnosis of the vital symptoms.

More specifically, phase contrast (PC) MRI sequences allow the acquisition of blood-flow data capturing both the velocity and the direction. If this phase contrast MRI is applied from multiple angles, a quantitative 3D blood-flow data set can be reconstructed. And if this process is repeated throughout a cardiac cycle, then a 4D blood-flow data set is obtained. The clear and efficient visualization of these data sets is one of the main challenges in this field.

In a previous work by Martijn Pieters a WebGL based interactive blood-flow animation by using lines as visualization technique was implemented. WebGL (Web Graphics Library) is a JavaScript API for rendering interactive 3D graphics within any compatible web browser without the use of additional plug-ins. The biggest strengths of WebGL are that it is accessible from any place worldwide with an internet connection and that most of the rendering can be done on the Graphic Card and thus, it is very fast even for big systems.

The *main contribution* of this work is the extension of Martijn Pieters web animation by implementing a particle-based visualization which has the following advantages over the line based visualization:

- Particles give a more realistic impression of the blood flow because of their resemblance to blood cells.

- They allow a better detection of abnormalities like vortices and deformed structures.
- The rendering of particles is much faster than the rendering of lines.
- A big number of particles can be drawn without creating a visual overload whereas lines are usually drawn on top of each other which results in a poor visualization.

2. Related Work

Medical Investigation of Flow Patterns: It has been observed in [vPBB*10], [vPBB*11], and [KGP*13] and by other authors that advanced MRI acquisition protocols enable in-vivo quantitative blood-flow information. This has fueled research in finding correlations between blood flow and the progression of cardiovascular diseases since pre-clinical research has indicated that atypical flow behavior directly relates to medical conditions. For example Cebral et al. [CMWP11] investigates the correlation between qualitative flow characteristics with respect to the risk of rupture for cerebral aneurysms.

Visual Exploration: Currently, there are visual exploration tools that assist users to get insight into the patient's hemodynamics. An example of these tools is the prototype presented by Martijn Pieters which is based on illustrative techniques like speed lines by van Pelt et al [vPBB*11] and Köhler et al. [KGP*13]. This prototype is based on the assumption that data is fully processed beforehand i.e. segmentation and pathline extraction have already been done and the data is ready for interactive visualization. A 4D flow data set consisting of 3D vector-valued data sets with different seeding times is used. Additional information like eigenvalues and velocities are also calculated beforehand. The methodology used to interactively visualize blood flow through a web browser is by the use of lines in WebGL. In the current work, we will follow a particle-based approach inspired by the visualizations in van Pelt et al. [vPBB*11].

Probing is done by use of mouse buttons which make viewpoint interaction very easy. Drag and drop is used to reposition the domain under visualization. Simple mouse operations are used to enlarge the object so that regions of interest are legible. Using the facilities made available in WebGL, visualization parameterization and time variation are achieved via drop-down menus and check boxes.

3. Explanation of the implementation

As stated before, the main contribution of this work is the actual implementation of the particle system in WebGL. To understand what was done, it is necessary to first understand how WebGL shaders work. Every object which is drawn in WebGL is using a vertex shader and a fragment shader. Generally spoken, the vertex shader can manipulate the attributes of vertices, such as color, size and time. On the other hand, the fragment shader takes care of how the pixels between the vertices look like. Usually, they are interpolated between the defined vertices following specific rules.

In the case of lines, the vertices are the preprocessed positions $p(t)$ at time t. The fragment shader is interpolating the time values for all pixels between two vertices and simply discards these pixels from drawing whose interpolated time is greater than the current time of the system. By continuously increasing the current time, the effect of an animation is created.

The drawing of particles is different. Here, the vertex shader is holding the current position of the particle and the fragment shader is just drawing the textures around the vertex. Therefore, it is necessary to know the exact position of each particle for each given time t. This is done by linearly interpolating the position from the closest time values in the preprocessed data (see Algorithm 1).

Data: Time **t**.
Result: Interpolated position $\mathbf{p}(\mathbf{t})$.
for *all particles* **do**
  Find t_{-1} and t_{+1}, the nearest time values before and after the given time t which is stored in the position data with corresponding positions $p(t_{-1})$ and $p(t_{+1})$.
  Linearly interpolate the current position as:

$$p(t) = \frac{p_{+1} - p_{-1}}{t_{+1} - t_{-1}} \cdot (t - t_{-1}) + p(t_{-1})$$

end

Algorithm 1: Pseudo-code for interpolating particle position

One problem is that these calculations cannot be done on the GPU because the position data is not available in the vertex shaders and cannot be transferred efficiently. In general, the GPU is very fast for parallelised computations, but rather slow for data transfer. Therefore, the interpolation has to run on the CPU and has to be optimized. The most expensive operation is the search of the nearest values t_{-1} and t_{+1}. But because the preproceseed position data is sorted, we can use a binary search which gives a satisfactory performance.

To draw particles with tails, i.e. with short speed-lines behind them, both the implementation of particles and lines was combined. One set of vertices is storing the current position of the particles and the other set is holding all the information about the lines. Then only those line segments are drawn which are close enough to the particle (i.e. which interpolated time is between 3-15ms behind the particle's time) and the effect of speed-lines is created. The length of the "tail" can be varied with the parameter tail length.

Motion Detection: In addition to the mouse-based probing that was used in the earlier prototype, we implemented a simple WebGL motion detector to rotate the 3D mesh of

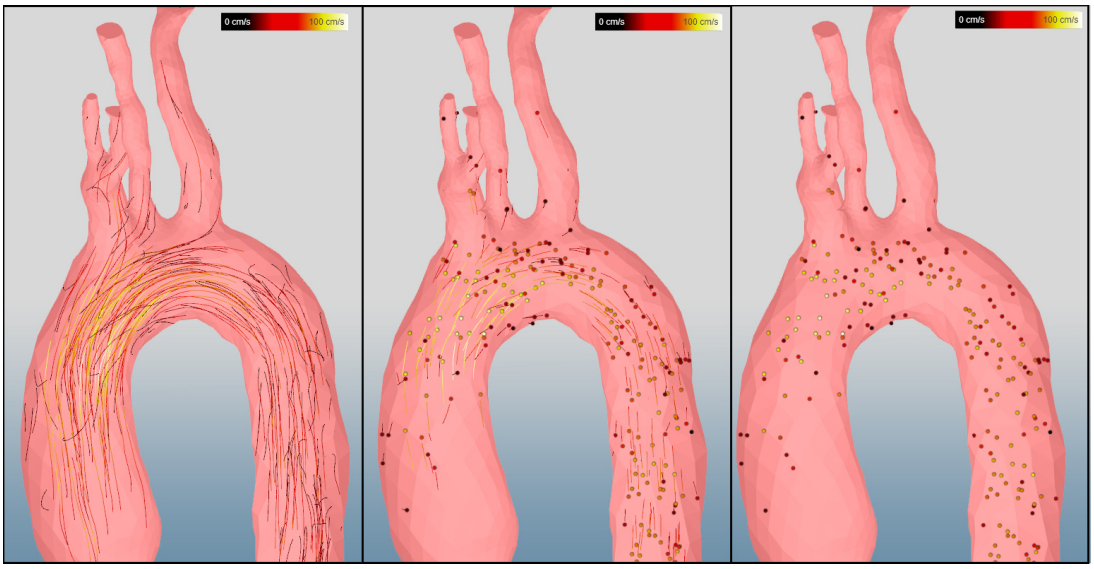

Figure 1: *Comparison of Different Visualization Techniques: (from left to right) lines, lines & particles, particles. Velocity is coded by different colors. As a static image, both the lines and the particles with speed-lines provide information about the blood -flow velocity and direction, whereas the particles lack the information of direction.*

the aorta. We use the inbuilt webcam which enables us to perform easy visualisation by object movement around the webcam. This may be handy in an environment that limits physical interaction with the computer. It is also good because you can move the mesh around when you are at a considerable distance from the computer. This is how motion detection works: Two frames of the video signal from the input (webcam) are compared. A small perturbation in position of the pixels is created by object movement in the vicinity of the webcam. The moving average of all pixel positions is computed. This moving average shows the average position of all movements on the screen. For better results, the lower part of the webcam display is ignored.

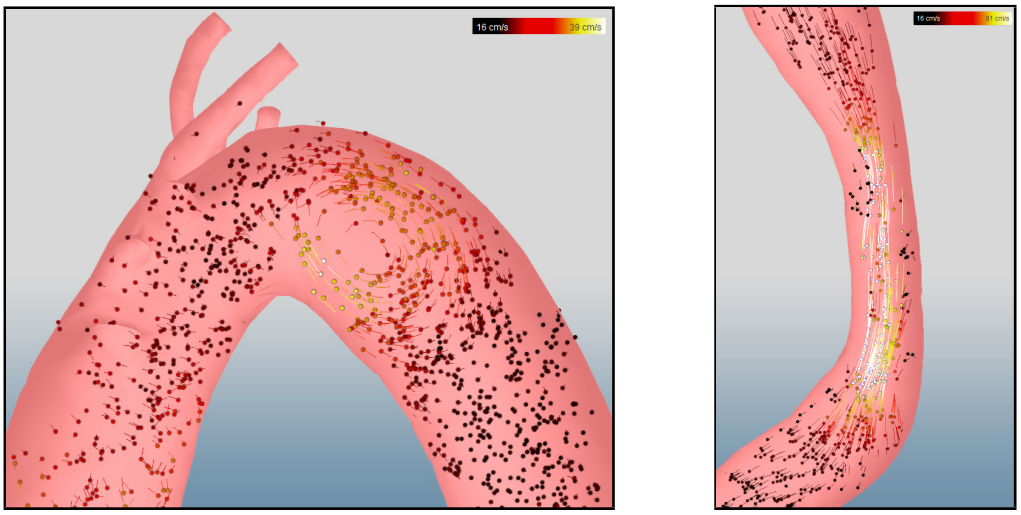

Figure 2: *Detection of a Vortex is shown on the left. On the right, we see part of the descending aorta with distorted shape (narrowed structure). Observe the difference in flow velocity along the narrowed structure.*

4. Results

Figure 1 shows the three different visualization techniques (lines, lines & particles, particles) which are now implemented in the web application. The velocity is coded by different colors. Figure 2 shows how a vortex and a narrowed structure can be detected by using particles with tails.

As a static image, both the lines and the particles with speed-lines provide information about the blood-flow velocity and direction, whereas the particles lack the information of direction. Therefore, the particles are not suitable for static images. However, particles are very well suited for animations in which they also capture the movement direction.

The advantage of particles over lines is that only the current position has to be handled, whereas for lines all vertices along the line have to be progressed. More precisely, there are 8466 lines with 224487 vertices in our first data-set (volunteer). If lines (or particles with lines) are used as visualization techniques, then all those vertices have to be sent to the GPU several times per second. This results in a very bad performance (3-10 FPS). But by using particles exclusively, only one vertex per particle has to be updated. It is then possible to draw all 8466 particles on an average computer with 40-60 FPS (maximum). This gives the opportunity for very illustrative real time web animations.

5. Conclusion and Future Work

An effective web based particle visualization was implemented in this work. It was shown that even complex data-sets can be animated in real-time within any compatible web browser to explore the Blood-Flow. In addition to that, it was shown that it is possible to use the available inbuilt computer resources to make fast and simple hands-off WebGL visualisation of blood-flow which may be useful in restrictive environments.

The biggest limitation of the current model is the poor performance when the whole data-set is selected and lines are included. A possible solution which is also implemented is a filter based on the eigenvalues (the so called λ_2 - Filter, see [JH95]) to filter out those lines which do not belong to a vortex. This effectively reduces the number of lines and results in a faster performance. However, it is also reducing the number of shown lines and therefore, it is hiding some useful information. A future project could be the development of a more efficient algorithm to draw the lines.

Another idea is the implementation of a Web based library consisting of different flow data sets of different parts of the human body. If this library is made freely accessible, then it would create a great tool for teaching and learning details about anatomy, blood-flow and related diseases for everyone who has an internet connection.

References

[CMWP11] CEBRAL J. R., MUT F., WEIR J., PUTMAN C. M.: Association of hemodynamic characteristics and cerebral aneurysm rupture. *Am J Neurorad 32*, 2 (2011), 264–270. 2

[JH95] JEONG J., HUSSAIN F.: On the identification of a vortex. *J. Fluid Mech. 285* (1995), 69–94. 4

[KGP*13] KÖHLER B., GASTEIGER R., PREIM U., THEISEL H., GUTBERLET M., PREIM B.: Semi-automatic vortex extraction in 4d pc-mri cardiac blood flow data using line predicates. In *IEEE Transactions on Visualization and Computer Graphics (Proc. IEEE Scientific Visualization)* (2013), vol. 19, pp. 2773–2782. 2

[NTS*12] NICHOLS M., TOWNSEND N., SCARBOROUGH P., LUENGO-FERNANDEZ R., LEAL J., GRAY A., RAYNER M.: *European Cardiovascular Disease Statistics 2012: European Heart Network*. Sophia Antipolis. Brussels and European Society of Cardiology, 2012. 1

[vPBB*10] VAN PELT R., BESCOS J. O., BREEUWER M., CLOUGH R. E., GRÖLLER E., TER HAAR ROMENIJ B., VILANOVA A.: Exploration of 4d mri blood-flow using stylistic visualization. In *IEEE Transactions on Visualization and Computer Graphics* (2010), vol. 16, pp. 1339–1347. 2

[vPBB*11] VAN PELT R., BESCOS J. O., BREEUWER M., CLOUGH R. E., GRÖLLER E., TER HAAR ROMENIJ B., VILANOVA A.: Interactive virtual probing of 4d mri blood-flow. In *IEEE Transactions on Visualization and Computer Graphics* (2011), vol. 17, pp. 2153–2162. 2