Scientific Visualization and Virtual Reality
Flow field visualization of Lattice Boltzmann grids.

Spiros Koulouzis <skoulouz@science.uva.nl>,
Grid Computing Master's Programme
University of Amsterdam

October 26, 2007

Abstract

In this assignment an effort is made to visualize Lattice Boltzmann data sets in VTK. More specifically, three data sets where used to extract and visualize vector fields, in order give a more comprehensive insight about fluid dynamics.

0.1. Introduction

The lattice Boltzmann method (LBM) is a powerful technique used for simulating fluid flow problems in complex geometries. It is a discrete computational method based on the Boltzmann equation. According to that, fluid is modeled by particles moving on a regular lattice, where each particle at a specific lattice point is assigned with a velocity according to a distribution function. The original Boltzmann equation describes the dynamics of the one-particle distribution function, $f(r, v, t)$. This gives the probability of finding a particle at the position r with velocity v at time t. Discretizing the previous function one gets: $f_i(r, t)$, where the velocity vector is mapped onto a discrete set of velocity vectors indexed by i. In other words this function is the probability of finding a particle at the discrete lattice site r at the discrete time t. So by using the discrete Boltzmann function, particles are moved to their next location, that is in a neighboring node in the lattice, also known as the propagation phase. Next the particles are assigned with their new velocity by also considering collisions. The results of simulating a a fluid flow with LBM need to be visualized, as this might provide more insight for the given problem, than simply numerical analyzing the data. Another advantage provided by visualization is that one may concentrate to a specific part of the simulation without loosing the general perspective. In this assignment three fluid flow simulations where visualized, a flow through a static mixing reactor (SMRX), a tornado and an artery while blood flows through it, using the Visualization Toolkit (VTK), in an effort to find the most suitable visualization for each simulation that would give the most information, using as less data as possible. VTK is an open source, image processing, and visualization toolkit developed in C++ under the influence of object-oriented principles. Several bindings may be used in order to develop code in Tcl/Tk, Java, and Python. Additionally VTK provides all the necessary tools for visualizing simulation data including scalar, vector, texture, and volumetric methods.

0.2. Fluid Flow And Visualization

As mentioned in the previous section, in this assignment three fluid simulations were visualized. The data used for this visualization are in a VTK format that consists of five parts, i) the header ii) the tile iii) the data type (ASCII or binary) iv) the geometry and v) the dataset attributes. In all three files the data (geometry) are represented as structured points, meaning that in all of them there is a vector field representing a particle's velocity at a specific position. Visualizing that vector field in a suitable way, will provide answers for specific problems.

0.2.1 Static Mixing Reactor

The static mixer, is a structure used in industry to mix extremely viscous fluids. It consists of specially designed stationary obstacles inserted in a pipe in order to mix fluids flowing through it. Its mixing mechanism relies mainly on redirecting the incoming fluid streams. Considering that, the visualization

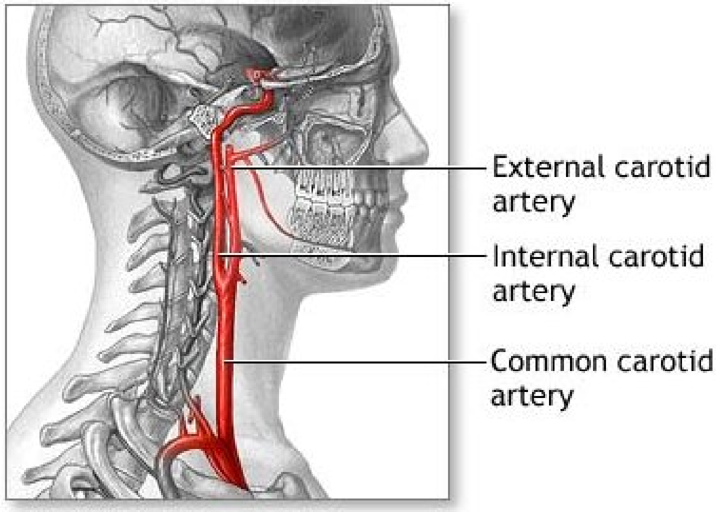

Figure 1. A representation of the carotid artery.

should clearly show the mixing of two separate fluids, one entering the reactor at the top half, and the other at the bottom half.

0.2.2 Tornado

Roughly speaking a tornado is caused when different temperatures and humidity meet to form thunderclouds. When a rising warm wind is strong enough to force itself up through a colder air layer, the cold air on top begins to move downwards, sending the rising warm wind spinning upward. The warm winds rotate faster and faster in a high column. In the center of this column is “the eye in the storm”, a region of mostly calm weather. If this is the case then using the appropriate visualization technique, the eye of the storm should be visible.

0.2.3 Carotid Artery

The common carotid artery shown in Figure 1 is an artery that supplies oxygenated blood to the head and neck. This artery divides in the neck (”bifurcates”) to form the external and internal carotid arteries. At this location, a buildup of ”atheroma” can occur that obstructs the bloodstream to the brain. Extracting the data from the provided data set, should reveal any anomalies in the bifurcation point in the course of one hart beat.

0.3. Implementation

In order to visualize the simulations described above, it is necessary to construct a visualization pipeline. A visualization pipeline, may be described as the process of turning data into a visual representation. This process is divided into distinct steps, where every step depends on the result of the previous. These steps are described below, while a pipeline may be seen in Figure 2

1. Data acquisition. Input data are gathered from some source, like MRI-scanners, some simulation, experiments, or some form of measurements in general.

2. Filtering. In this stage of the pipeline, data are filtered to more relevant, sets for the problem at hand, for example extracing a scened scull from an MRI. In other words in this stage one chooses what it wants to visualize,

3. Mapping. The filered data are mapped into geometric primitives, like points, lines, etc. At this point there are a number of primitives suitable for visualizing the focused data. This posses a trade-off between expressiveness and effectiveness.

4. Rendering. Having all the necessary primitives it is now time to transform them into images. Before turning the geometric primitives into images, they are assigned with visual properties, like color, opacity, or shadow. The goal here is not photorealism, but rather information conten.

0.3.1 VTK and QT

In the context of VTK the visualization pipeline consists of data and process objects. Data object are there to represent, create, access and delete information. Additionally data objects may offer metadata method such as minimum or maximum data values. Intuitively process objects acts on data objects, or creates new ones. Filters or mappers, belong to the process objects. Hence it is clear that VTK adopts an object oriented approach to the visualization pipeline.

The approach taken in this assignment is first extract the data from the provided file, and the determin the best way to visualize the vector filed Figure 3. This is done with C++ because visualizing complex data sets is a CPU demanding task. Thus C++ was chosen in hope for better performance.

Although VTK provides all the necessary tools for developing scientific visualizations, it lacks on GUI tools. QT is a GUI API, that enables easy and fast development of interfaces. For that reason QT was used to provide interactivity and control over the application.

0.3.2 Static Mixing Reactor

As described in the previous section the first stage in creating a visualization pipeline is to acquire the raw data. Likely data are provied in the form of VTK file formats. The next step is to extract the "relevant" data. In this case the reactor, together with the vector vieled. Extracting the reactor is a trivial task. Obtaining a data reader, and applying a contour filter provides the structure of the reactor in the render window. The next step is to visualize the vector field in an appropriate way. Since there is not a unique representation for clearly showing all the aspects of the mixing reactor, the application provides, with the help of QT, a number choices of viewing the two fluids passin through the reactor. These might be streamlines that are colored according to their velocity, streamtubes the thickness of which reviles the velocity, or glyphs the size and orientation of which show not only the velocity but also the direcon of the fluid.

0.3.3 Tornado

Visualizing a tornado simulation, dose not involve extarcing any object, but making the vector field more comprehensive is a bit more challenging. The nature of this simulation involves visualizing vectors that are situated in a column and each one has a different magnitude and direction of the velocity. Again the user is free to choose the representation the is suitale for the problem, as in the case of the SMRX, with the addition of ribbon that show vector's "twist" in the tornado.

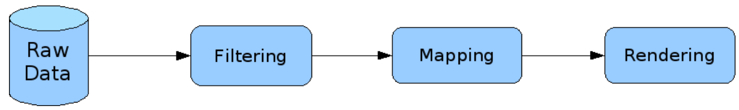

Figure 2. A visualization pipeline.

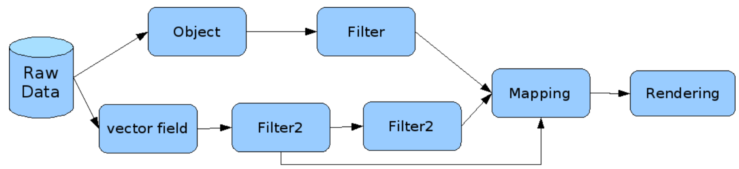

Figure 3. The approach taken in visualizing the vector fields together with an object.

0.3.4 Carotid Artery

The carotid artery, flows more or less the same philosophy as the SMRX, with the difference that the data provide a cycle of a heartbeat, that needs to be animated. First of all the vector filed needs to be read before running the animation, otherwise , the animation would not be smooth. Second in this representation glyphs are the most suitable choice, as they would have a liter effect on the animating.

0.4. Results

0.4.1 Static Mixing Reactor

The goal of this visualization is to determin if two separate fluids are mixing in the other end of the reactor. But before determining this, it would be useful to understand how a fluid reacts while passing through the reactor. Figure 4 and 5, clearly show that while inside the reactor, the velocity of the fluid increases.

Looking at Figure 6 and 7 one could see that the two fluids (represented with different colors) are indeed mixing in the other end of the reactor, but not as effectively as expected/.

0.4.2 Tornado

Visualizing the “eye of the storm” is better achieved with the use of arrow glyphs, since they don’t only provied a comprehensive structure of the tornado, they also clearly show the “eye”. Figures 8 and 9 illustrates this point.

0.4.3 Carotid Artery

Since the point of intrest in this visualization is the bifurcation in the carotid artery, glyphs are the most useful representation. They reveal the direction of the vector in that point. Observing Figure 10 one could see that in the external carotid artery some of the bloodstream is heading opposite of the flow. This could be evidence of atheroma, although this is not the area of expertise of this writer. Another useful representation for the carotid artery, is the one shown in Figure 11, where the whole vector filed is shown.

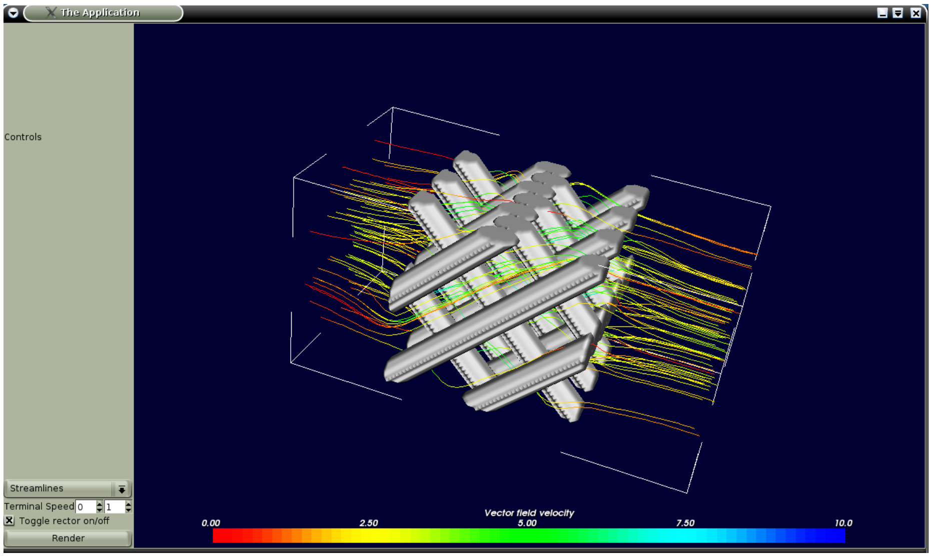

Figure 4. A fluid, represented in streamlines passing through a static mixing reactor.

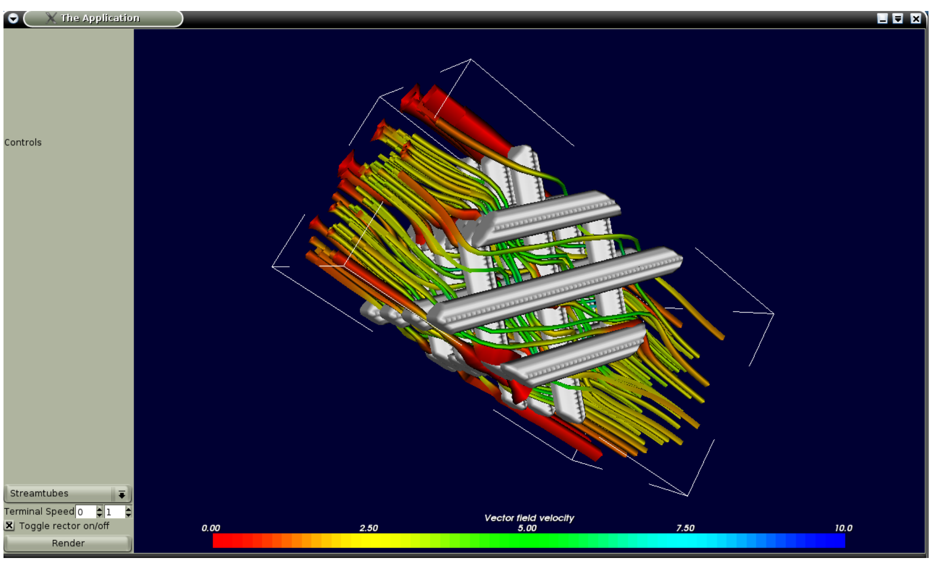

Figure 5. A fluid, represented in streamtubes passing through a static mixing reactor.

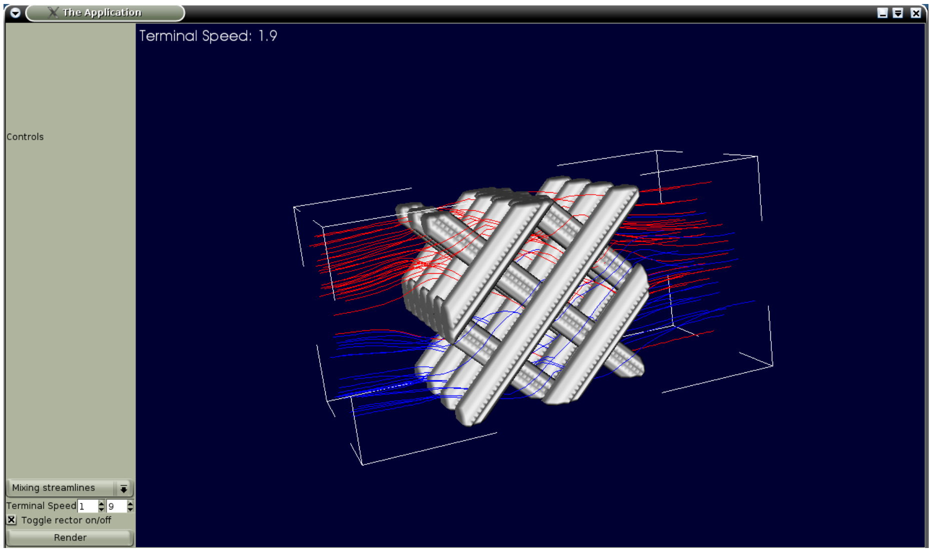

Figure 6. Two fluids, passing through a static mixing reactor.

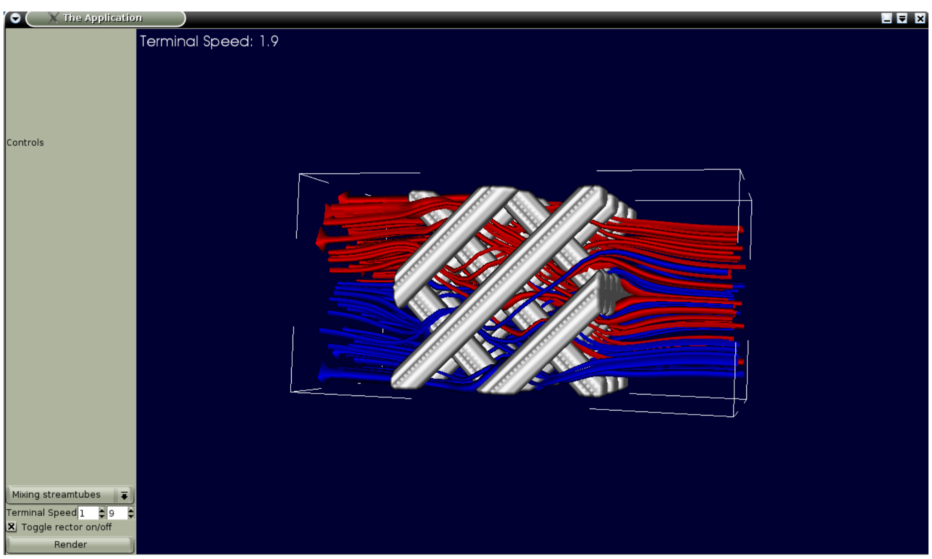

Figure 7. Two fluids, represented in streamtubes passing through a static mixing reactor.

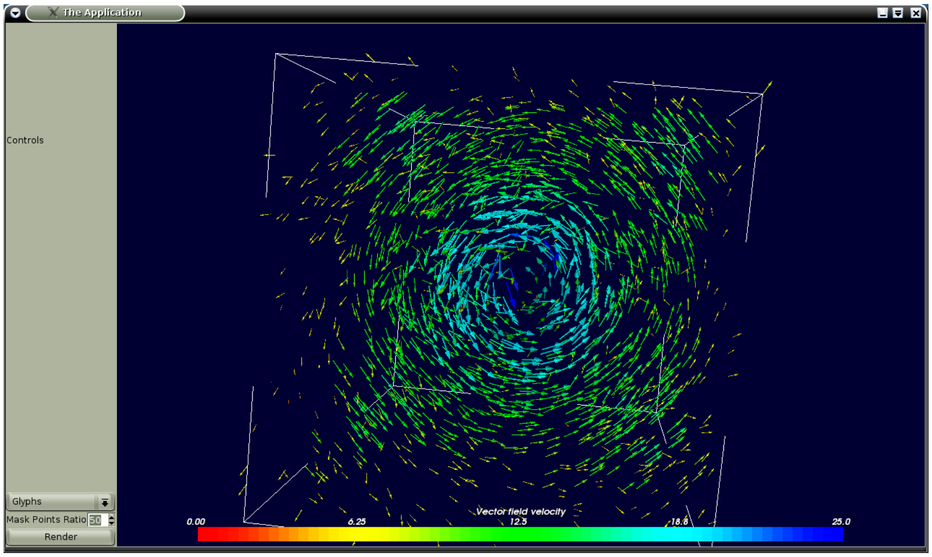

Figure 8. The “eye of the storm. It is obvious that in the center of the tornado around the eye wall there is a clam area.

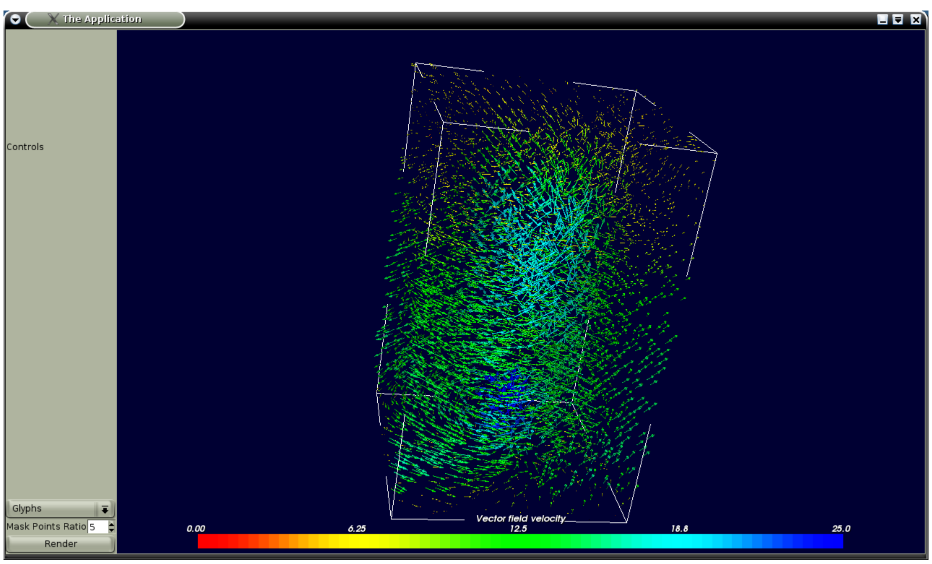

Figure 9. The tornado simulation. In this Figure the structure of the tornado is showing in a more comprehensive form.

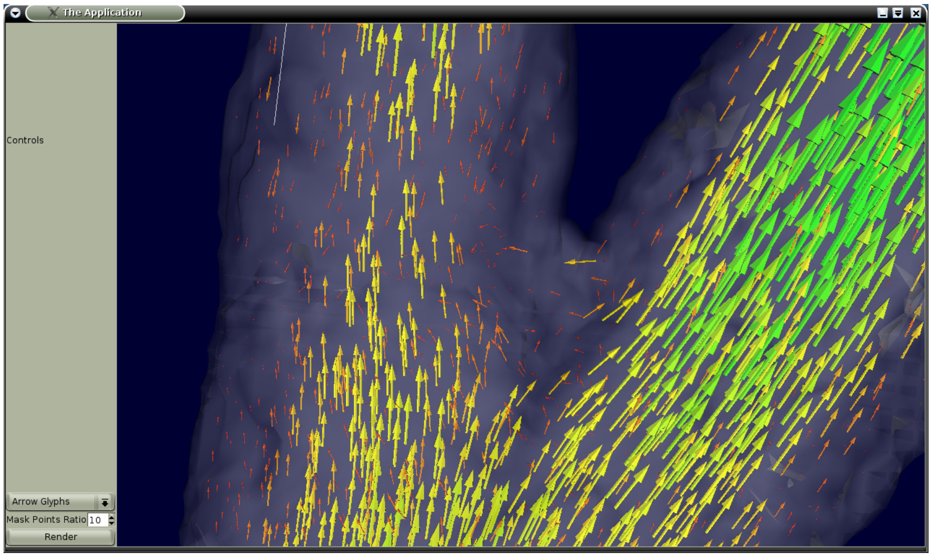

Figure 10. The carotid artery in the bifurcation point. It seems that some of the bloodstream is heading oppsite the flow.

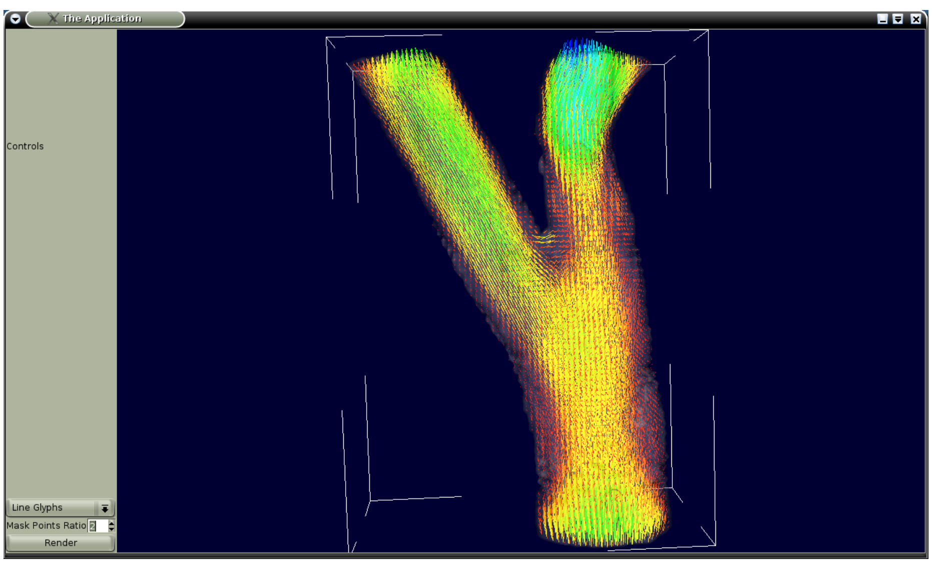

Figure 11. The carotid artery with whole of the vector filed.

0.5. Conclusions

Scientific visualization is all about looking at massive datasets and giving them a meaning. This meaning depends on the problem at hand, and where the focus should be. In this assignment where the all the datasets were Lattice Boltzmann simulations, streamlines and glyphs proofed to be the most informative form of visualization. The don't only provide magnitude but also direction of a vector. This form of visualization, thou would not be scruffily applied in an MRI scan where the goal is to detect anomalies in the scull. VTK is the most established toolkit for scientific visualization applications, as it provides an easy and effective way for representing scientific data.