Interactive Exploration in Virtual Environments

Belleman, R.G.

Publication date
2003

Chapter 6

Integrating all: simulated vascular reconstruction in a virtual operating theatre*

"Training for minimal invasive surgery, or "trying to get the most gut for the cut", has its limitation due to the lack of repetitive training opportunities on either animals or humans."

Newsbytes News Network, 1997.

6.1 Introduction

In this chapter we describe an interactive dynamic exploration environment (IDEE) that combines simulation and interactive visualization in virtual reality (VR) into an environment that enables pre-operative surgical planning of vascular reconstruction procedures. This environment uses many of the ideas described in the previous chapters. It shows how the simulation and interactive visualization execute on distributed systems, communicating with each other over a high throughput network interface. Visualization of the simulation results and interaction with the environment take place from within a virtual environment (VE). The simulation runs on a parallel system for best performance.

6.1.1 Abdominal vascular reconstruction

Vascular disorders in general fall into two categories: *stenosis*, a constriction or narrowing of the artery by the buildup over time of fat, cholesterol and other substances in the vascular wall, and *aneurysms*, a ballooning-out of the wall of an artery, vein

*Parts of this chapter have been published in *R.G. Belleman and P.M.A. Sloot. "Simulated vascular reconstruction in a virtual operating theatre", number 1230 in* Excerpta Medica, International Congress Series, pages 938–944, 2001.

or the heart due to weakening of the wall [15, 34]. Aneurysms are often caused or aggravated by high blood pressure. They are not always life-threatening, but serious consequences can result if one bursts.
A vascular disorder can be detected by several imaging techniques such as X-ray angiography, MRI (magnetic resonance imaging) or computed tomography (CT). Magnetic resonance angiography (MRA) has excited the interest of many physicians working in cardiovascular disease because of its ability to non-invasively visualize vascular disease. Its potential to replace conventional X-ray angiography that uses iodinated contrast has been recognized for many years, and this interest has been stimulated by the current emphasis on cost containment, outpatient evaluation, and minimally invasive diagnosis and therapy [266].
A surgeon may decide on different treatments in different circumstances and on different occasions but all these treatments aim to improve the blood flow of the affected area . Common options include *thrombolysis* where a blood clot dissolving drug is injected into, or adjacent to, the affected area using a catheter; *balloon angioplasty* and *stent placement* which is used to widen a narrowed vessel by means of a inflatable balloon or supporting framework; or *vascular surgery*. A surgeon resorts to vascular surgery when less invasive treatments are unavailable. In *endarterectomy* the surgeon opens the artery to remove plaque buildup in the affected areas. In vascular bypass operations, the diseased artery is shunted using a graft or a healthy vein harvested from the arm or leg.
The purpose of vascular reconstruction is to redirect and augment blood flow, or perhaps repair a weakened or aneurysmal vessel through a surgical procedure. Although the optimal procedure is often obvious, this is not always the case; for example, in a patient with complicated or multi-level disease. Pre-operative surgical planning will allow evaluation of different procedures *a priori*, under various physiologic states such as rest and exercise, thereby increasing the chances of a positive outcome for the patient [234].

6.1.2 What is needed?

The test case described in section 6.1.1 contains all aspects of an interactive dynamic exploration environment. Our aim is to provide a surgeon with an environment in which he/she can try out a number of different bypass operations and see the influence of these bypasses. The environment needs the following:

- An environment that shows the patient under investigation with his affliction. A patient's medical scan is 3D, so to obtain best understanding on the nature of the problem, the surgeon should be able to look at his specific patient data in 3D, using unambiguous visualization methods. We have described the Virtual Radiology Explorer (VRE) environment that allows for the visualization of 3D medical scans in section 2.3 (page 29).

- An environment that allows the surgeon to *plan* a surgical procedure. Again, this environment should allow interaction in a 3D environment. For example,

the CAVE environment allows us to interact with 3D computer generated images using 6 degrees of freedom (DOF) interaction devices [52, 205]. We have described an architecture that allows interactive manipulation of virtual 3D objects in a virtual environment in section 4.2 (page 71). Note that *visual* realism is not the primary goal here; what is more important here is *physical* realism, and then only of particular features of fluid flow, as discussed later[†].

- An environment that shows the surgeon the effect of his planned surgical procedure. As the aim of the procedure is to improve the blood flow to the affected area, the surgeon must have some means to compare the flow of blood before and after the planned procedure. This requires the following:
  - A *simulation* environment that calculates the important properties of blood flowing through a patient's vascular system (i.e. pressure, velocity, wall shear stress). We will describe this simulation environment and the methods for obtaining the input data for this environment from patient specific data in section 6.2.
  - A *visualization* environment that presents the results of the simulation in a clear and unambiguous manner. We have described a scientific visualization environment that can be used in virtual environments in section 4.2 (page 71).
  - An *exploration* environment that allows the researcher to inspect and probe (qualitatively and quantitatively) the results of the simulation (e.g. means for performing measurements, annotate observations, releasing tracer particles in the blood stream, etc.). We have described an architecture that allows measurements to be taken in a virtual environment in section 4.5 (page 86). An environment that visualizes the results of blood flow simulation will be described in section 6.3.

6.2 The lattice Boltzmann method for flow simulation

The lattice Boltzmann method (LBM) is a mesoscopic approach for simulating fluid flow based on the kinetic Boltzmann equation. In this method fluid is modeled as particles moving on a regular lattice. At each time step particles propagate to neighboring lattice points and re-distribute their velocities in a local collision phase. This inherent spatial and temporal locality of the update rules makes this method ideal for parallel computing [123]. During recent years, LBM has been successfully used for simulating many complex fluid-dynamical problems, such as suspension flows, multi-phase flows, and fluid flow in porous media [134]. All these problems are quite difficult to simulate by conventional methods [122, 124].

[†]This in contrast to research projects toward virtual operating theatres that include the simulation of tissue deformation and realistic blood spills [12, 19].

The data structures required by LBM (cartesian grids) bear a great resemblance to the grids that come out of CT and MRI scanners. As a result, the amount of preprocessing can be kept to a minimum which reduces the risk of introducing errors due to data structure conversions. In addition, LBM has the benefit over other fluid flow simulation methods that flow around (or through) irregular geometries (like a vascular structure) can be simulated relatively simply. Yet another advantage of LBM is the possibility to calculate the shear stress on the arteries directly from the densities of the particle distributions [8]. This may be beneficial in cases where we want to interfere with the simulation while the velocity and the stress field are still developing, thus supporting fast data-updating given a proposed change in simulation parameters from the interaction modules.

6.2.1 Performance of the parallel LBM flow simulation kernel

Figure 6.1 illustrates the performance increase that is achieved by using a parallelized implementation of the flow simulation kernel on three different parallel computing systems;

- a 128 node SGI Origin 2000 system, each node consisting of a MIPS R12000 processor running at 300 MHz, running IRIX; the processors are interconnected through a ccNUMA (cache coherent Non Uniform Memory Access) architecture;
- a 20 node cluster-of-workstations (COW), each node consisting of 2 symmetric multi-processing (SMP) Intel Pentium-II processors running at 300 MHz (i.e. the system contains 40 processors in total), running Solaris, interconnected by Myrinet;
- a 40 node cluster-of-workstations (COW), each node consisting of an AMD Athlon processor running at 700 MHz, running Linux, interconnected with a dedicated fast (100 Mb/s) switched ethernet.

Figure 6.2 shows the speedup of the parallel implementation. Although the execution time per iteration was higher on the SGI system, this figure shows that the code runs at higher efficiency. This can be largely attributed to the more efficient communication architecture that is used in the SGI system. The figure also shows erratic speedup behaviour on the 20-node COW when more than 20 processes are used, even though the system has 40 processors. This can be explained from the fact that each node contains two processors but one network interface; when more than 20 processes are scheduled on the system, some nodes will run multiple processes on the same node, scheduled over the two available processors. When communication takes place, the processors compete for the single network interface, thereby inhibiting speedup. Because the test case environment's architecture uses the distributed pipeline architecture described in section 5.2.2 and because the simulation component is, in general, the slowest executing component, the performance of the simulation environment in total is greatly increased. In the current LBM simulation environment the execution time of the lattice Boltzmann kernel is independent of the morphology through or

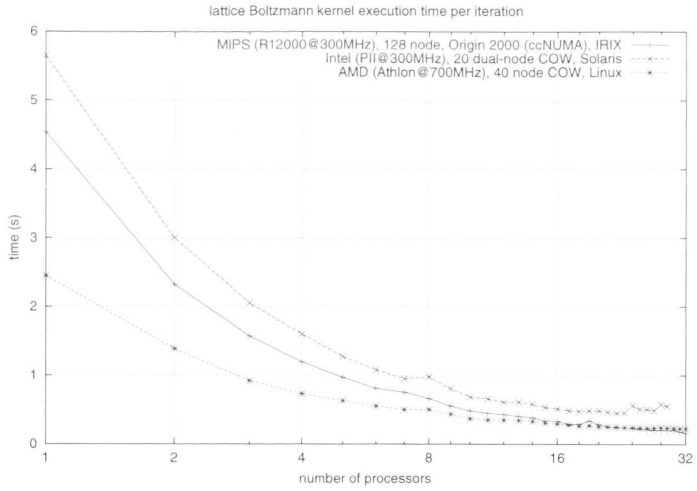

Figure 6.1: Iteration time of the parallel lattice Boltzmann kernel on different multiprocessor systems.

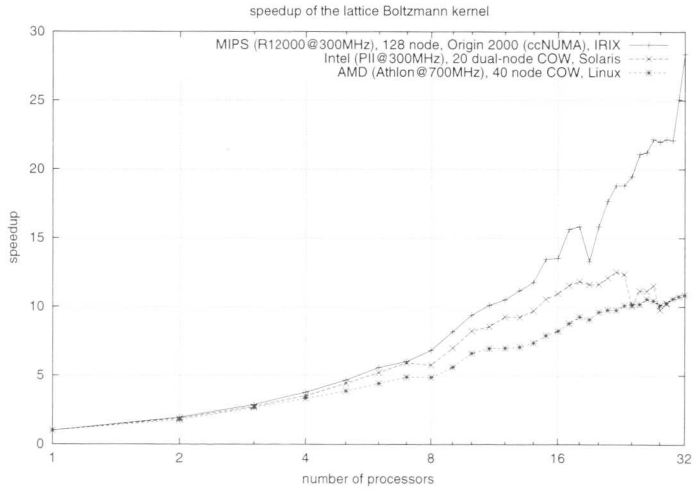

Figure 6.2: Speedup of the parallel lattice Boltzmann kernel on different multiprocessor systems.

around which the flow is calculated (given equal grid dimensions and simulation parameters). However, the ratio of fluid nodes and solid nodes can make an important difference. Especially in angio-vascular applications, the number of solid nodes is often much higher than the number of fluid nodes. An optimization can be applied in

the lattice Boltzmann kernel to speed up the simulation by disregarding solid nodes in the propagation phase.

6.2.2 LBM grid generation

For our purpose, we consider lattice Boltzmann grids to consist of isotropically structured nodes with one of the following five properties (see also Figure 6.3):

1. Fluid nodes define the geometry through which flow is simulated.
2. Solid nodes delimit the domain of fluid nodes, and as such define the "shape" of the geometry around (or through) which the fluid flow is simulated.
3. Boundary nodes are solid nodes that border fluid nodes. They are often defined as "first order bounce back nodes" or as "second order no-slip nodes" to describe the behaviour of the flow at the boundary of a solid and the fluid. What exactly constitutes a "border" depends on the type of neighbourhood that is used in the LBM algorithm; in 2D, the D2Q5 and D2Q9 models are often used (respectively with a 4 and 8 boundary neighbourhood); in 3D, the D3Q19 model is very popular (26 boundary neighbourhood).
4. Inlet nodes define the initial distribution of forces from where the flow enters the system. Often the inlet flow profile is statically defined to approximate a system in which the flow evolves under the influence of a time-continuous velocity profile. In dynamic systems, the inlet velocity profile is changed dynamically over time to approximate the forces that are exerted on the system by an external force (such as the pulsating flow as a result of a beating heart cycle, consisting of alternating cycles of *systoles* where blood is pumped out of the heart and *diastoles* where the heart is in rest).
5. Outlet nodes define the behaviour of the flow as it exits the system. Common conditions on the outlet nodes are "free flow", where the forces traveling out from the system encounter no resistance, "periodic flow", where the forces are wrapped back into the inlet nodes simulating a closed, recursive, system and "restrictive flow" in which a situation is approximated where flow encounters a high resistance (such as flow through small branches and capillaries that are too small to be accurately simulated using LBM).

LBM requires that the input grids comply to the following set of rules:

- inlet nodes may not neighbour boundary nodes, only fluid nodes,
- outlet nodes may not neighbour boundary nodes, only fluid nodes,
- solid nodes may only neighbour boundary nodes,
- boundary nodes may only neighbour fluid nodes.

The purpose of automatic LBM grid generation is to construct grids that comply to this set of rules.

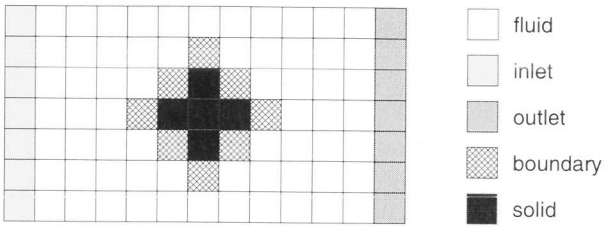

Figure 6.3: Different node types in an example D2Q5 lattice Boltzmann grid.

6.2.3 LBM grid generation from medical data sets

As mentioned earlier, the basic structure of the grids used in LBM bear great resemblance to the medical scans obtained from a patient by medical imaging devices such as CT and/or MRI scanners. In general these scanners produce stacks of two dimensional images that, when stacked together, form a three dimensional image volume. For the purpose of flow simulation it is imperative that the structures through which the flow must be simulated are isolated from the raw medical scan as accurate as possible. This process is known as "image segmentation".

Image segmentation

Image segmentation, like flow simulation, is an area of active scientific research [9, 76, 148]. Many techniques exist, ranging from manual techniques where pixels are tagged by experienced radiologists with a thorough understanding of human anatomy, to fully automated techniques. All techniques rely on the basis that a sufficient signal-to-noise (S/N) ratio is present in the images so that the relationship of pixels with others that belong to the same structure can be identified. Obtaining images with a sufficiently high S/N ratio begins in the radiology department, at the time the scan is made.

In the case of vascular structures, the imaging process is referred as "angiography" (from the Greek *angeion*, "vessel" and *graphien*, "to write or record"). Radiologists are trained with the knowledge on the physics behind the different medical scanners and therefore know how to obtain good quality images. One currently used method is computerized tomographic angiography (CT angiography or CTA). Here, a high contrast CT scan is produced using intravenous water-soluble iodinated agents to image the vascular system. Time-of-flight magnetic resonance angiography (TOF-MRA) is a non-invasive imaging technique that is well suited for obtaining high S/N ratio scans. Although TOF-MRA does not require a contrast agent, it has the disadvantage that the acquisition times are usually high (in the range of 10-30 minutes for high resolution scans). This may hamper the maximum attainable resolution in regions of the body that move under the influence of respiratory motion.

Provided that sufficient contrast is present in the medical scans, the raw data from the medical scanner is first segmented so that only the arterial structures of inter-

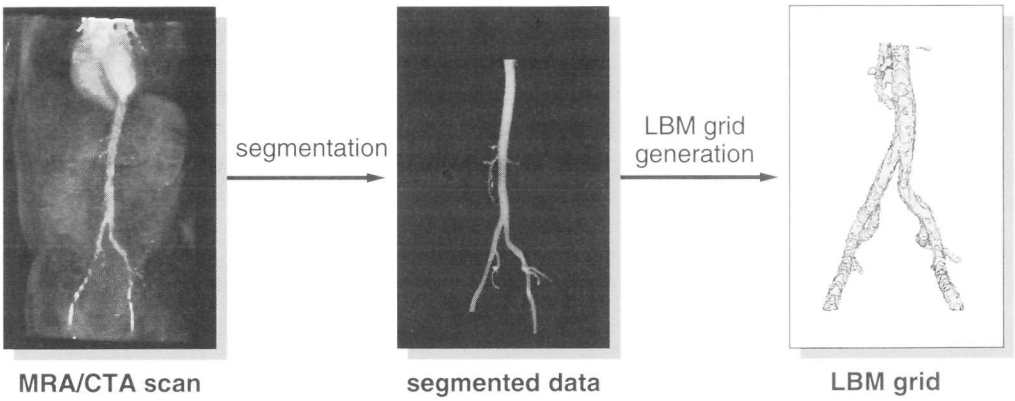

Figure 6.4: LBM grids are generated from raw medical scans through a combination of segmentation and image processing techniques.

est remain in the data set (see also Figure 6.4). In some rare cases, the S/N ratio may be sufficiently high that a level-threshold segmentation technique is suitable to segment the structure of interest from the raw medical scan. If this is not possible, more sophisticated techniques are required, such as for example the wave propagation technique developed at the Leiden University Medical Center [206].

Automatic generation of boundary nodes

The segmented data set is then converted into a grid that can be used in LBM; boundary nodes, inlet nodes and outlet nodes are added to the grid using a variety of image processing techniques. The implementation of the Lattice Boltzmann used requires that grids are isotropic. This means that the settings of the medical scanner either have to be such that isotropic voxels are produced or, otherwise, the image has to be resampled onto an isotropic grid. Conventional image processing techniques can be applied here, provided that valid interpolation algorithms are used as determined by the acquisition method that was used.
Boundary nodes are generated using a morphological image operation known as image dilation (see Figure 6.5). The effect of this operation is that one layer of nodes is generated around the segmented image that conforms to the requirements of the LBM simulation. By subtracting the original segmented image from the dilated image, an image is obtained that contains the boundary nodes only.

6.2.4 LBM grid generation and editing from polygonal data

As our aim is to provide physicians with an environment in which vascular reconstruction procedures can be simulated, methods should be available by which a reconstructive procedure can in some way be approximated by a LBM grid. We limit

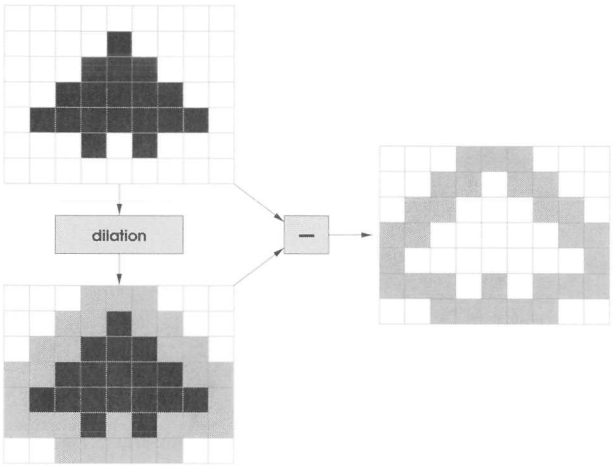

Figure 6.5: Boundary node generation using image dilation.

ourselves to reconstructive procedures that alter vascular geometry, more specifically, the types of treatment that are performed with stenotic and aneurysmal disease as described in section 6.1.1, with the exception of thrombolysis using chemical agents.

Distance sampling

A suitable method for generating grids from simple curves is distance sampling. Here, an implicit model is created from the input geometry by computing the smallest unsigned distance from the input geometry to each voxel in a grid. Figure 6.6 shows the execution pipeline that has been implemented to do this. The size of the output grid and the number of points in the input curve should be defined in such a way that a sufficiently accurate representation can be obtained.
The distance sampling algorithm visits each node in the output grid and records the

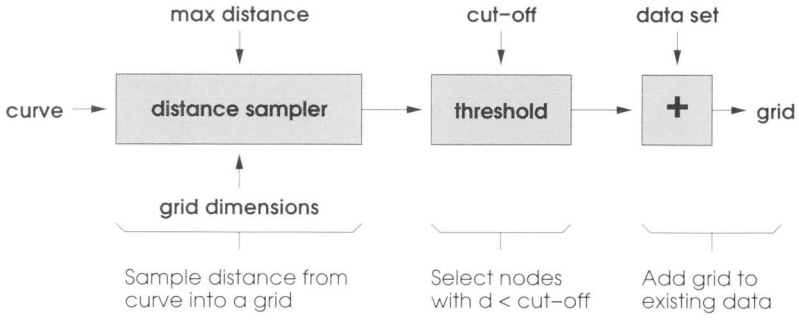

Figure 6.6: Grid generation using distance sampling.

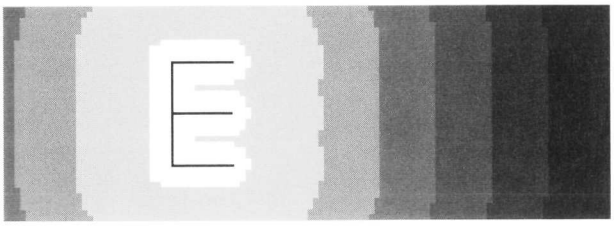

Figure 6.7: Distance sampling from a line representation of the letter 'E' onto a 3-bit deep two dimensional image. Larger distance is represented by a darker colour.

minimal distance of this node to each of the points in the input curve. When this distance is represented by a colour, we obtain the picture shown in Figure 6.7. To decrease the execution time of the distance sampling algorithm, especially in the case of large grids, a maximum distance parameter limits the traversal to grid nodes that are within a specified distance from the input curve. The distances values stored in the sampled grid can be used to extract grid nodes with a distance less than a threshold value. This threshold value defines the diameter of the sampled curve; higher values result in wider structures while lower values in structures that are narrower. Finally, the thresholded grid is added to the medical data set for which the procedure it to be simulated.

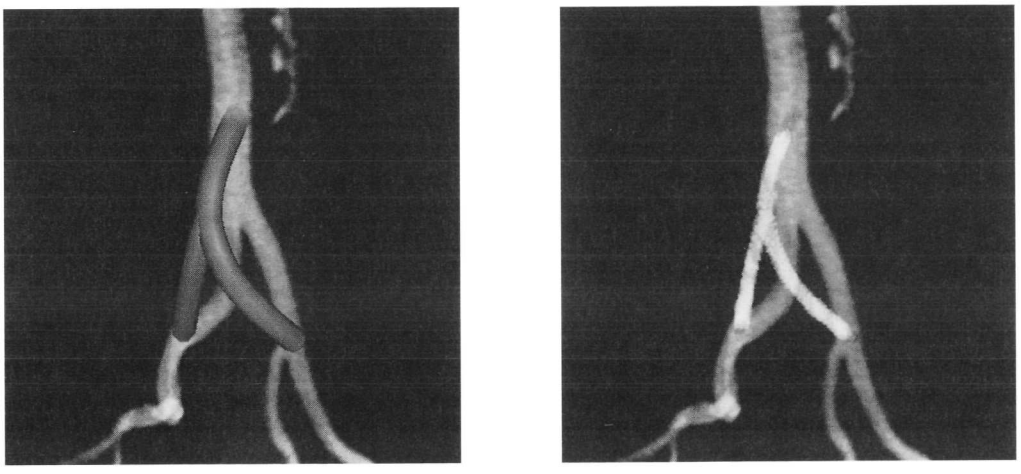

Figure 6.8: Example application of distance sampling to approximate a vascular reconstructive procedure used in abdominal aortic aneurysm repair known as a "aorto-femoral bypass graft with proximal end to side anastomosis" [233]. The picture on the left shows a spline representation of the bypass placed onto the aorta and both iliacs. The picture on the right shows a distance sampling of the bypass, added to the original data set.

Figure 6.8 shows an example of how the distance sampling technique can be used to approximate the placement of a bypass known as an "aorto-femoral bypass graft with proximal end to side anastomosis" [233]. This procedure is applied in patients with aneurysms that involve the aorta and the iliac arteries as well.

Grid generation and editing using stencils

The distance sampling method is well suited to approximate vascular reconstruction procedures that can be described by simple curves, such as shunts or bypasses with a constant diameter. Shapes that are more complex, however, can not be easily created in this manner.
Stencils are patterns that are used to mask (copy or replace) pixels in an image in location where pixels in the stencil are set. In its most simple form a stencil is an image by itself. The previously described distance sampling technique can be used to generate stencils from other shapes that are not images. To change an image, the stencil pattern is transformed to the desired location of the image. Next, each voxel in that image that overlaps a set voxel in the stencil is either replaced or copied to respectively delete or add pixels in the image. Figure 6.9 illustrates this process with a 2D stencil on a 2D image; the process is identical for 3D stencils and images.

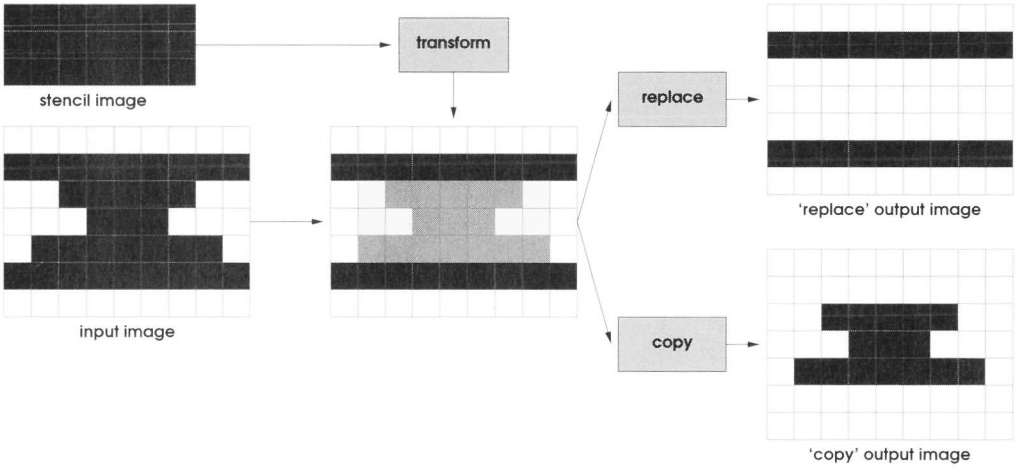

Figure 6.9: The stencil operator used to copy or replace regions of an image.

Grid generation using implicit functions

Implicit functions are functions that, given a 3D coordinate (x,y,z) coordinate, evaluate to a value $w = f_v(x,y,z)$ and a gradient vector $\vec{v} = f_g(x,y,z)$. A carefully structured complex polygon structure, such as those modeled using computer aided design (CAD) or 3D modeling packages, can be used as an implicit function by determining, for each

3D voxel in an output grid; (a) the distance from the voxel to the nearest polygon in the model and (b) the vector to this polygon. This Alias|Wavefront model of an airplane was sampled onto a grid of $64 \times 64 \times 64$ voxels using an implicit function operator in Vtk.

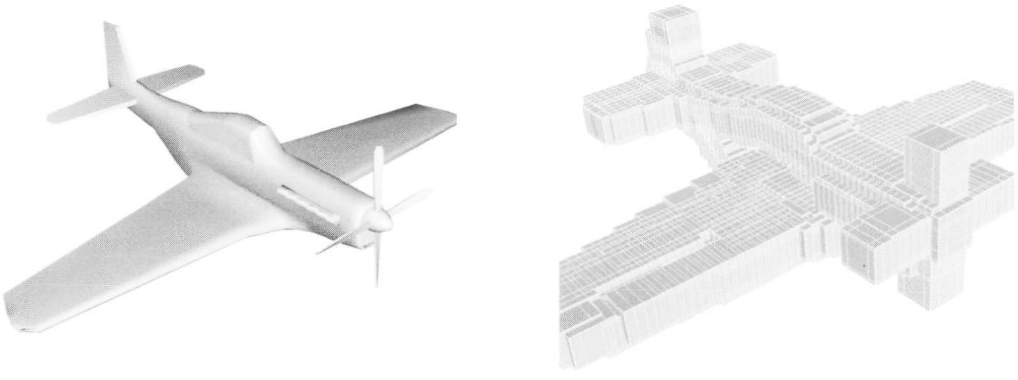

Figure 6.10: A model of an airplane sampled onto a grid of $64 \times 64 \times 64$ *voxels using an implicit function.*

6.2.5 Interactive LBM grid editing in a VE

Surgical procedures in a VE are simulated using a combination of the 3D manipulation techniques offered by SCAVI, described in section 4.2, and the grid generation techniques described in the previous sections. This combination allows a user to interactively add and/or delete areas in the LBM grid corresponding to the procedure that is simulated. Similar grid generation techniques as described above are used to ensure the grids comply to the demands imposed by LBM. Figure 6.11 shows an example of the interactive placement of a simulated bypass, represented by a spline. SCAVI is used to allow the user to manipulate the spline and its control points, represented by spheres, to set the start and end point and the path of the spline. Once the bypass is in place, a distance sampling method is used to sample the bypass and add it onto the existing grid, as previously illustrated in Figure 6.8. After adding the boundary nodes this grid can be converted to a new LBM grid which can be used as the input for a new flow simulation.

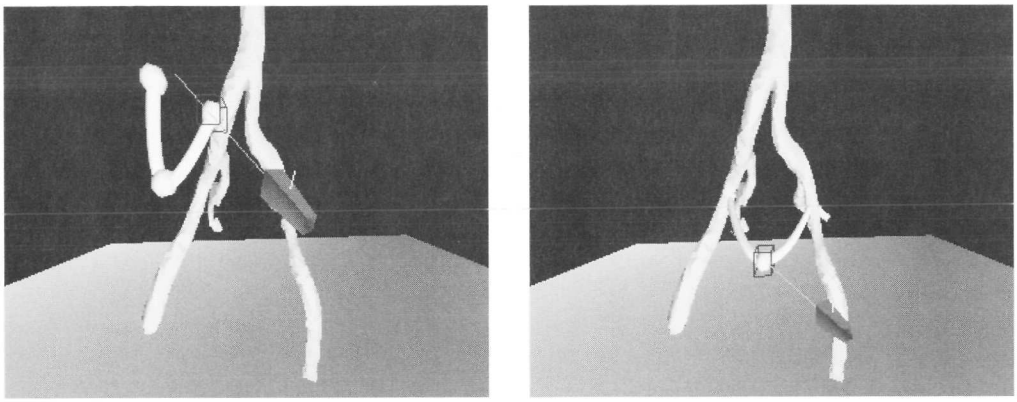

Figure 6.11: Interactive placement of a bypass, represented by a spline. The start and end point of the spline and its path is adjusted using the control points represented by spheres.

6.3 Interactive 3D flow visualization in VEs

The visualization of flow fields is difficult. The challenge in the visualization of flow fields (or vector fields in general) lies in the mapping of vectors to comprehensible visual constructs. Various methods have been designed that do this [56, 57, 185, 246]. The effectiveness of a vector field visualization method depends on the spatial and temporal complexity of the underlying vector field. Although very effective visualization methods have been developed for the representation of 2D vector fields, these methods are not always suitable for 3D fields. The reason for this is often that the 2D visualization methods lack the depth cues that make them suitable for 3D vector field visualization. One way through which these depth cues can be added is through the use of advanced graphics rendering techniques (depth of field, shadows) and VR display and interaction technology (in particular stereoscopic projection and head motion parallax) .

For the purpose of our test-case, we are concerned with the visualization of flow properties that help a surgeon in making a decision on a vascular treatment and deciding what the effect is of a proposed treatment on blood flow [225]. The flow properties that should be represented at minimum are flow direction, flow speed and pressure. We have designed and implemented a number of flow visualization methods for use in interactive VEs [193]. These methods make use of vector field visualization methods in Vtk. The interaction with the flow visualization methods is handled by SCAVI (see section 4.2). The flow visualization methods can be categorized into global and local visualization methods.

6.3.1 Global flow visualization methods

Global visualization methods display the overall behaviour of the flow and are represented by isosurface modeling on scalar properties of the flow (like pressure and flow speed) or by a glyph representation. Glyph visualization of flowfields is a technique where a shape is positioned at each vector, oriented in the direction of the vector and (if desired) coloured by an additional scalar property of the flow at the vector location. The shapes that are most often used in this case have length and direction, such as arrows, so that the shape of the flow becomes apparent. This technique is quite effective for 2D vector fields. For 3D fields, the number of vectors quickly clutters the view on the global flow domain. This can be compensated by subsampling the input data set, but in this case care should be taken of undersampling, in which case important information could be missed, and aliasing effects in the case of regular sampling. Aliasing effects can be reduced by applying a random sampling of the vector field.
With a large number of vectors in the vector field, the number of objects that need to be rendered also increases which may impose a significant load on the rendering engine and, as a consequence, lead to a reduced frame rate. This problem can be solved by the use of glyph objects with as few geometric primitives as possible, such as a simple hook consisting of two lines[‡].

6.3.2 Local flow visualization methods

A common method to visualize flow fields is by tracing the path of a virtual particle through the velocity vectors. Stepwise integration of a particle at position $x(t)$ and velocity $v(x(t))$ at time t through a velocity field is obtained by repeated application of the following procedure:

$$x(t+\delta t) = x(t) + \int_{t}^{t+\delta t} v(x(t))dt$$

The integration is normally done using a second or fourth order Runge-Kutta method. A path trace is computed starting from an initial location $x(0)$ and repeated application of this procedure with a constant or variable time step. The procedure stops when (1) the virtual particle leaves the velocity field, (2) the speed of the virtual particle drops below a predefined threshold, or (3) a predefined number of integration steps have been performed. After this, the positions computed by this procedure can be used to draw a curve through the velocity field. This method can be repeated for more particles, each at a different starting location. For example; by using a set of equidistant start locations on a line segment (called a "rake"), a similar representation can be obtained as with the smoke released from such an instrument used in real wind tunnel experiments.
Although this visualization method produces acceptable results when applied to 2D velocity fields, it quickly creates occluded views when applied to 3D fields. This can

[‡]Note that lines can not be shaded by most rendering engines, thereby removing a depth-cue which may be important in understanding a 3D vector field.

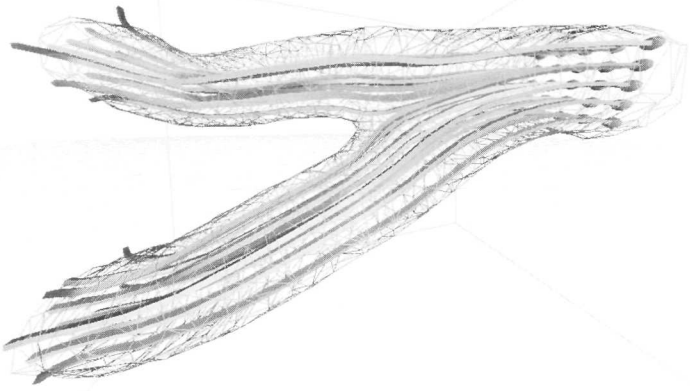

Figure 6.12: Interactive 3D flow visualization in a virtual environment (image created by Hans Ragas). See also colour reproduction on the back cover.

only be partially resolved through depth cues such as stereoscopic projections, motion parallax and shading. Furthermore, it is not easy to tune the integration parameters in such a way that clear results can be obtained.

6.3.3 Results

Figure 6.12 shows an example of an interactive 3D flow visualization method used in a VE. This example shows a streamline representation of a flow field that was simulated through a bifurcated abdominal aorta using the lattice Boltzmann method. The input data set for this simulation was generated from a magnetic resonance angiography (MRA) scan of a healthy patient. This data set was converted into a lattice Boltzmann grid using the methods described in section 6.2.3. The streamlines in this visualization are generated from an interactively moveable plane-shaped source shown at the right of the picture. The colour of the streamlines represents flow speed (from blue for low velocity to red for high velocity). The size, shape and number of particles in the source is interactively controlled by the user from within the environment. The visualization shows that the flow speed is low at the artery walls and higher just after the bifuraction.

6.4 Summary and conclusions

Using the design decisions on the construction of an interactive dynamic exploration environment described in the previous chapters and the additional methods described in this chapter, we have been able to construct an environment that combines fluid flow simulation, visualization and interactive exploration in a virtual environment for the purpose of simulated vascular reconstruction. The environment allows users

to simulate flow through complex shaped geometries and explore the simulation results at run-time. The interactive visualization of results from the simulation in a 3D virtual environment provides insights that would have been difficult to obtain on conventional desktop environments. Furthermore, the interaction capabilities allow the 3D geometry of the geometry to be changed from within the environment to simulate a vascular reconstruction procedure. These changes can then be given back to the fluid flow solver to see the effect of these changes on the flow properties (see also Figure 6.13). Although the system we have developed is by no means ready for use in realistic situations, we have shown that our design choices result in an environment with great potential.

Figure 6.13: The effect of a change in geometry on a flow field (see also colour reproduction on the back cover).

There are however also still several areas that require further attention. Although the 3D flow visualization environment provides an extensive array of possibilities to explore a fluid flow domain, the parameterization of the visualization methods is problematic. An attempt has been made to automatically derive the best parameters for each visualization algorithm, but these will have to be addressed. Also; the current interaction methods to simulate vascular procedures bear little or no resemblance to the actual procedures performed by vascular surgeons. If a system as proposed in this chapter is to be used in realistic situations, the interaction methods will need to be thoroughly reviewed together with prospective users. This process has recently been initiated at the Section Computational Science.