Sketch-based Dynamic Illustration of Fluid Systems

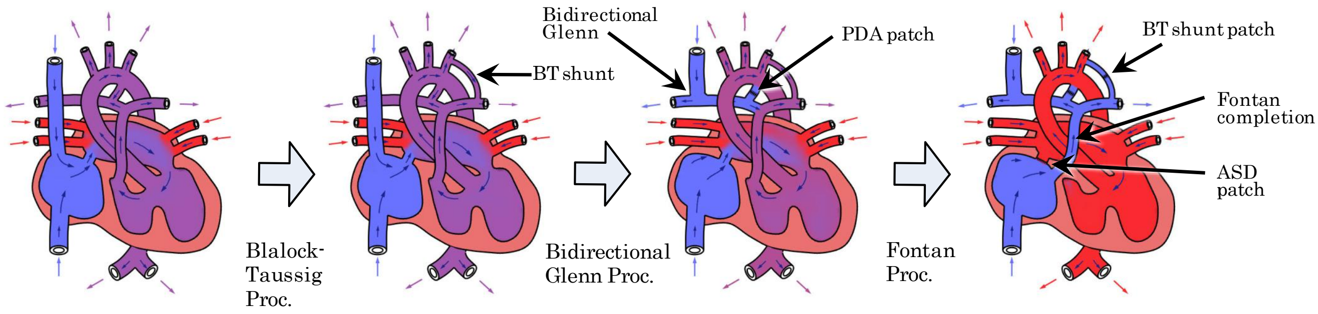

Figure 1: *Illustrations created using our system to describe the surgical repair procedure of tricuspid atresia (TA). The user interactively edits the illustration and the system continuously presents the corresponding blood flow computed by simplified fluid simulation.*

Abstract

This paper presents a lightweight sketching system that enables interactive illustration of complex fluid systems. Users can sketch on a 2.5-dimensional (2.5D) canvas to design the shapes and connections of a fluid circuit. These input sketches are automatically analyzed and abstracted into a hydraulic graph, and a new hybrid fluid model is used in the background to enhance the illustrations. The system provides rich simple operations for users to edit the fluid system incrementally, and the new internal flow patterns can be simulated in real time. Our system is used to illustrate various fluid systems in medicine, biology, and engineering. We asked professional medical doctors to try our system and obtained positive feedback from them.

CR Categories: I.3.8 [Computer Graphics]: Applications—; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques;

Keywords: dynamic illustration, real-time fluid simulation, sketch interface

1 Introduction

Fluid systems are ubiquitous. A typical fluid system includes a fluid to carry the materials to be transported, pipes and regions to distribute the fluid, and external sources and sinks to drive the flow through the system. In the fields of medicine, biology, and engineering, it is important to be able to illustrate how these systems work and how to operate on them dynamically. For example, a doctor may need to illustrate how blood flow patterns inside normal and abnormal hearts differ and how a defective heart can be repaired with a series of surgical operations; a biologist may need to explain how an animal circulatory system functions and how it transports materials; an engineer may need to explain to a customer how air circulates inside a house after installing new air conditioners. Even though these kinds of illustrations can be enhanced using simple fluid simulations and visualizations, standard simulation systems are too complicated for use in casual and interactive discussions.

This paper presents a sketching system that incorporates a background fluid simulation for illustrating dynamic fluid systems. Our method combines sketching, simulation, and control techniques in one user interface and can produce illustrations of complex fluid systems in real time. Users design the structure of the fluid system using basic sketch operations on a canvas and progressively edit it to show how flow patterns change. The system automatically detects and corrects the structural errors of flow simulation as the user sketches. A fluid simulation runs constantly in the background to enhance flow and material distribution in physically plausible ways.

We developed a hybrid fluid simulation method using multilayered two-dimensional (2D) space for the background simulation. We chose not to use three-dimensional (3D) simulation because this makes the simulation difficult to control and visualize. Two-dimensional illustrations are easier to create and understand, and are therefore more widely used for depicting fluid systems in medical and engineering education. We developed a hybrid method because standard 2D hydrodynamics simulations are too slow for our purposes and cannot handle the multilayered structures typical in these illustrations. Our method combines a hydraulic network model and a multilayered hydrodynamics model to enable efficient flow simulation on different levels. We used a hydraulic graph to represent the entire system on a coarse level and mapped each fluid region onto a multilayered grid to calculate local flow. The flow in the hydraulic network is calculated first and is then used to drive the hydrodynamics model in various regions by providing boundary velocities. The structures inside regions can conversely influence the flow in the network to avoid topological errors in model coupling.

The main contributions are listed as follows:

- We propose a novel application: editable, dynamic fluid illustration for interactive discussion and communication of fluid systems.
- We present a novel 2.5D representation for such fluid illustrations consisting of regions and pipes, and present sketch-based user interfaces to edit them interactively.
- We present a novel hybrid algorithm for fluid simulation that couples hydraulic simulation for the global flow in a network with hydrodynamics simulation for regional fluid.
- We show the feasibility and effectiveness of the system by presenting a solid implementation with feedkback from professional medical doctors.

2 Related Work

Explanatory illustration is an effective way to communicate scientific and technical information visually. Many studies have focused on how to generate these illustrations automatically [Ebert et al. 2005]. Recently, researchers have been especially interested in generating illustrations to explain dynamic physics systems. Davis [2007] proposed a 2D physics illustrator to describe body interactions in a constraint system. Mitra et al. [2010] proposed a method based on shape and contact analysis to illustrate how mechanical assemblies function. Vainio et al. [2005] designed a virtual learning environment for explaining complex medical phenomena to medical students. Zhang et al. [2006] proposed a system for designing vector fields. Researchers have also proposed various kinds of visualization techniques to depict time-varying flow velocity fields, including streamlines [Zöckler et al. 1996] and advecting textures [Cabral and Leedom 1993; van Wijk 2002]; see [McLouglin et al. 2010] for a survey. These techniques rely on offline simulation results and can provide accurate visual effects. However, preparing these kinds of illustrations is time consuming, and users cannot interactively edit the fluid system under consideration. Our system fills a gap in interactive explanatory fluid illustration by providing an editable illustration tool based on a sketching interface and real-time simulation.

Sketching interfaces have been used for designing static 3D scenes, such as models of geometrical objects [Igarashi et al. 1999; Gingold et al. 2009], scene phototypes [Zeleznik et al. 1996], and plants [Ijiri et al. 2006]. They have been combined with animation systems to illustrate dynamic phenomena [Davis et al. 2007; Davis 2007; Davis et al. 2008] including fluid phenomena [Angelidis et al. 2006; Okabe et al. 2009]. Certain physics engines such as Crayon Physics [Purho 2008], Phun [Ernerfeldt 2008], and Physicafe [Prometech 2008], also provide sketch interfaces that allow users to design their own physics systems. However, these systems are mainly designed to simulate fluid flow in an open space and are not directly applicable to flow in a closed circuit as typically found in medical and engineering illustrations.

Fluid simulation is an established technique widely used in many fields. Many researchers have worked to produce photorealistic 3D fluid effects in real-time applications [Müller et al. 2003; Treuille et al. 2006; Cohen et al. 2010]. In addition to these 3D simulation techniques, 2D hydraulic models are also widely used to enable efficient simulation of system behavior on a macroscopic level. Yu et al. [2009] used a hydrographic network model to simulate rivers in real time. Sewall et al. [2010] used a graph model to simulate traffic flow in a city. Hydraulic models are also widely used in biomedical modeling, especially blood flow simulation [Formaggia and Veneziani 2003; Nobile 2009; Almeder 1999]. Hybrid methods have been proposed to model both macroscopic and microscopic fluid phenomena in one scene. Irving et al. [2006] coupled 2D and 3D simulation techniques to model large quantities of water with surface details. Nobile [2009] proposed a two-way coupled method to simulate blood flow at various resolutions and tested the system using a very simplified model. In contrast, our model couples the global hydraulic network and local hydrodynamics regions in one-way to provide plausible real-time enhancement for illustrations. Similar approach is used in human body simulation coupling local muscles and global skeletons [Lee et al. 2009].

3 User Interface

3.1 Overview

The system presents a fluid circuit with several basic elements: pipe network, region, source, and sink. The pipe network connects the different parts of a fluid system and allows materials to be transported inside it. It has both spatial and topological structures, and can be represented by a hydraulic graph. Regions are large flowable areas such as the heart in a blood circulatory system. These regions have different shapes and both static and dynamic inner structures. These structures substantially influence the flow patterns inside the regions. Sources and sinks drive the fluid system by providing external flows to the system. Our sketching interface provides users with various simple tools to help them design and edit these basic elements. Users can edit the locations, shapes, and connectivities of regions and pipes using geometry tools; change the layer configuration using the layer tool; control the flow using the flow control tool; and edit the temporal behavior of objects inside regions using the dynamic object tool. A flow simulation constantly runs in the background to provide physically plausible results for these operations in real time.

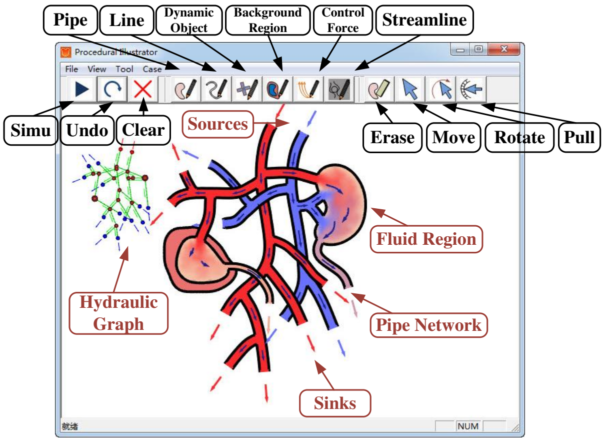

Figure 2: *System screenshot.*

3.2 Geometry Editing

Users design a fluid system with two basic geometry tools, ”line” and ”pipe” as shown in Figure 3top. Users draw contours to describe fluid domains and their inner structures using the ”line” tool. We provide three different drawing styles: drawing a freeform stroke by dragging, drawing a rectangle by rubber banding, and drawing a polyline by multiple clicks. Users draw pipes of different diameters using the ”pipe” tool, and link the different fluid domains or create pipe networks. Materials are transported in the designed system composed of regions and pipes based on hydraulic rules. Users can draw accessory regions using the ”background region” tool. An accessory region is only for illustration purposes and does not affect the simulation. Users can delete a region or pipe using the ”eraser” tool. We provide edit tools for users to modify the elements on the canvas. Users can move regions and pipes using the ”move” tool, rotate them using the ”rotate” tool, and deform them using the ”pull” tool as shown in Figure 3middle [Igarashi et al. 2005]. An end point of a moved or deformed pipe is automatically connected to a nearby pipe or region. These operations can be combined to represent complex actions such as surgical operations as shown in Figure 1.

3.3 Layer Operation

Medical illustration involves rigorous restrictions of layer configurations; the local stacking sequence of vessels and organs must be the same as in real anatomy. To meet this requirement, our system enables layering operations by assigning depth values for objects on a 2.5D canvas. Each primitive (triangles for regions and segments for pipes) is assigned an integral depth value s. These depth

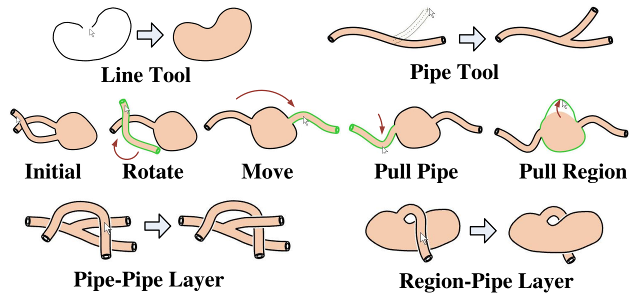

Figure 3: *Basic operations. Upper: Line tool (left) and pipe tool (right). Middle: rotate, move and pull operations. Bottom: Layer Operations.*

values are automatically adjusted in local areas to represent inter-occlusions and self-occlusions.

Users edit the local occlusion sequences of regions and pipes using a popup menu selection such as "bottom pipe to top" or "top pipe to bottom" to obtain the desired stacking effect as shown in Figure 3bottom. We implemented a simplified local layering system similar to [McCann and Pollard 2009] to help users swap the stacking order in local areas. The layer sequences of the overlapping pipes and regions are stored in the local areas and are adjusted when users click on the overlapping parts. Using these layer operations, users can create various illustrations that strictly follow the conventions used in medicine as shown in Figure 10.

3.4 Flow Control

The system provides various methods for controlling the flow inside the system. When a user draws a pipe, the starting point becomes the inflow node and the end node becomes an outflow node. Users can change the inflow or outflow conditions, inflow material color, and inflow velocity of a node with a popup menu operation on a pipe end. Users block and unblock a pipe with a popup menu operation on a pipe as shown in Figure 4top. Users can also add flow sources inside a region using a "source" tool. They can sketch a shape, translate it, and rotate it to achieve different source effects as shown in Figure 4middle. The source adds velocity forces around it and produces a material for material transportation. This feature is useful for planning the layout of flow sources inside a space, such as planning the locations and directions of indoor air conditioners as shown in Figure 11c.

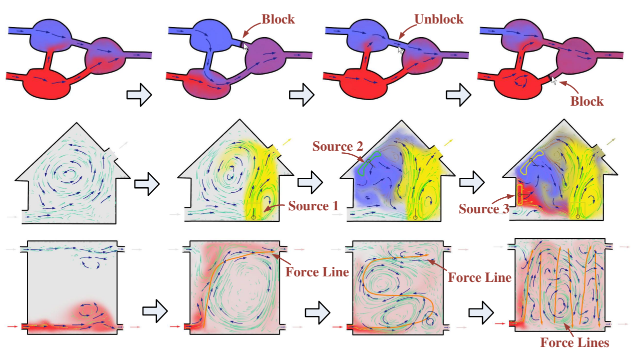

Figure 4: *Flow control operations. Upper: block and unblock pipes. Middle: adding flow sources inside regions. Bottom: adding control force lines (orange).*

Our system provides a "control force" tool that enables users to control flow patterns inside regions as shown in Figure 4bottom. Users draw force lines using the tool inside a region and the system adds smoothed control forces to the fluid around it before solving the regional Navier-Stokes equations. This will generate a new flow field according to a user's particular illustration needs.

3.5 Dynamic Object

For a dynamic fluid system, users might want to add moving objects inside it such as a piston moving inside an engine. We provide simple sketch tools that enable users to add dynamic objects with periodic actions. As shown in Figure 5, users draw a new dynamic object by drawing the outline with a dynamic object tool. An anchor point appears at the center of the object. Users change the location of the anchor point by clicking within the object. Users define the linear movement by sketching a path starting from the anchor point using the move tool, and define the angular movement by rotating the object using the rotate tool. The objects then move and rotate periodically with a predefined speed according to these inputs. The movement of these dynamic objects affect the fluid around it by exerting impulse boundary forces to satisfy the moving solid conditions.

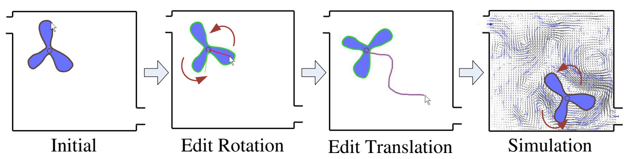

Figure 5: *Edit the temporal actions of dynamic objects.*

4 Algorithm

This section describes the simulation algorithm running in the background to enhance the illustration. As shown in Figure 6, the basic workflow of the algorithm contains hydraulic graph construction with structural error detection, a flow simulation coupling network and regions, and material transportation. When user sketch on the canvas, a hydraulic graph is incrementally built to represent the fluid circuit on a high level. This graph is used to calculate the flow in the circuit by solving a linear system based on hydraulic rules. A multilayered Euler solver driven by this network flow runs to calculate the flow patterns inside local fluid regions. Materials with different colors are transported in the fluid system to enhance the illustration based on the flow solved by the hybrid solver.

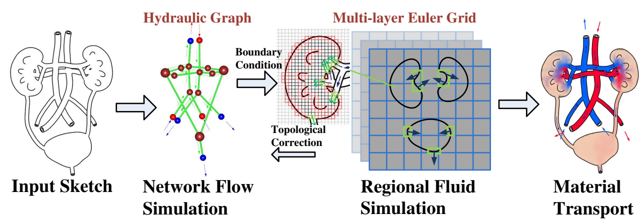

Figure 6: *Basic workflow of the system.*

4.1 Hydraulic Graph

When a user sketches on the canvas, a hydraulic graph is incrementally constructed at the same time by identifying new regions and pipe elements. This graph represents the spatial and topological relationships between regions and pipes and helps to calculate the network flow in the system.

Two basic graph elements, nodes and edges, are abstracted from the input sketches. Nodes contain both regions and joints. Each sketched closed contour is identified as a region. Start points, end points, and joints of pipes are treated as joints. Each sketched pipe is treated as a directed edge connecting regions and joints. Actual flow directions can be opposite to the edge direction, in which case, the flow velocity would be negative. We use $G(n_n, n_e)$ to represent a graph containing n_n nodes and n_e edges. To describe the graph topology, a node-edge matrix $\mathbf{M}$ is used to represent the relationships between nodes and edges in the graph. For $G(n_n, n_e)$, $\mathbf{M}$ has n_n rows and n_e columns. The value of element M_{ij} equals $+1$ when edge j starts from node i, -1 when edge j ends at node i, and 0 otherwise.

As shown in Figure 7, hydraulic nodes can be categorized into four different flow types: *source*, *sink*, *saddle*, and *isolated*. Fluid flows through source nodes into the system, through saddle nodes to different parts, and through sink nodes to the outside. There is no flow in isolated nodes. Source and sink nodes are distinguished intuitively as users sketch: when a pipe is drawn from a blank area, its start point is a source node; when a pipe ends in a blank area, its end point is a sink node.

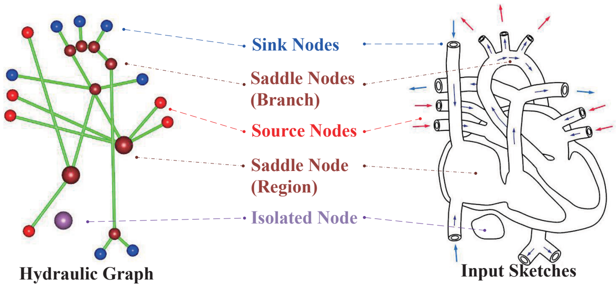

Figure 7: *Four different types of hydraulic nodes.*

When sketching a fluid system, users may often draw structures that are invalid for solving the network flow, e.g., a network without a source or sink, or a pipe with blocked nodes. Our system eliminates these invalid parts on both the global and the local levels. We detect input errors by maintaining and validating all of the connected subgraphs when a hydraulic graph is updated. These invalid subgraphs are cut from the graph being sent to the network solver, and the flows in these subgraphs are set to zero.

Errors also occur locally when users draw obstacles inside a region. When a region is completely blocked or separated into several different parts by these obstacles, the network solver is not aware of it and still treats it as a flowable node, which causes errors in the coupled network and region simulation. We use a node split and merge algorithm to solve this problem. When users draw obstacles inside a region, we employ a flood fill algorithm on its fluid cells to test its connectivity. When a region is subdivided into several smaller subregions, new subnodes are created and the connected pipes are re-linked to these subnodes. The new network flow is calculated based on the updated graph and the new boundary conditions are set for the regional solver as shown in Figure 8. In turn, when some blocks are removed inside a split node, subnodes will be merged in the graph to reflect these changes.

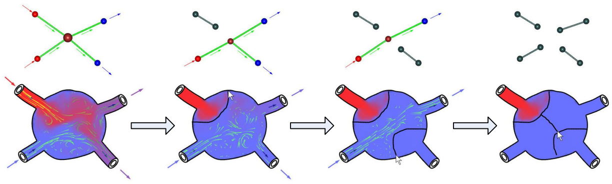

Figure 8: *Regional node split.*

4.2 Hybrid Flow Simulation

We propose a new hybrid model that couples network flow and regional flow simulations to provide physically plausible results at different levels. The model calculates the flow direction and rate of each pipe, and depicts the velocity field of each regional node. We use the numerical human artery model developed by Almender [1999] to solve the flow in the hydraulic network, and a multilayered 2D Navier-Stokes solver to solve the flow in regional nodes. The network model drives the flows inside local domains by providing boundary conditions to the regional solver. Note that this is a one-way process. The flow inside a local domain including those caused by sources and moving obstacles in it does not influence the network flow. We chose this one-way model because it is much easier to compute and control than a two-way model [Nobile 2009]. This can be problematic if our goal was an accurate simulation; however, fast computation and controllability are more important for interactive creation of explanatory illustrations.

We model fluid transportation through the flow network based on the hydraulic graph constructed in Sec.4.1. To calculate network flow, we use the method proposed by Almender [1999], which is described briefly in the Appendix. This involves solving a linear system with two variables, the mean flow velocity of each pipe and the hydraulic pressure of each node, based on the relationships between network morphology and hydraulics.

The multilayered regional fluid solver models the velocity field in the regional nodes driven by the network inflow and outflow conditions. Each fluid domain is mapped onto a sliced 2D Euler grid to run hydrodynamic simulation. We current do not support self-intersecting regions. The system solves the Navier-Stokes equations on the grid:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = -\frac{1}{\rho}\nabla p + \mathbf{g} + \nu \nabla \cdot \nabla \mathbf{u}, \tag{1}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{2}$$

in which p is pressure, u is velocity, ρ is fluid density and ν is viscosity.

We solve Equations 1 and 2 using the standard 2D finite difference method [Bridson 2008], which is widely used in graphics applications. The solver is driven by the pipe flow velocities as its boundary conditions. The velocity value of each boundary cell is calculated as $\|v_p\| \cdot \mathbf{n}_p$, in which $\|v_p\|$ is the flow velocity of the connected pipe and $\mathbf{n}_p$ is the direction of its outlet. As in a standard Euler solver, we split the Navier-Stokes equations into three simpler equations and solve them separately: $\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} = 0$ (advection), $\frac{\partial \mathbf{u}}{\partial t} = \mathbf{g}$ (volume force), and $\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho}\nabla p$ with $\nabla \cdot \mathbf{u} = 0$ (incompressibility). An extra cell-sorting step is needed before the incompressibility step to ensure the correctness of applying cells on different layers in a standard 2D solver. All cells in the computation domain are sorted by the index values in the sequence of dimension x, y, and s. The sorted cell list is then used to construct the sparse symmetric positive definite matrix to solve the Poisson equation of pressure in the incompressibility step.

We solve the dynamic objects inside regions using the immersed boundary (IB) method [Mittal and Iaccarino 2005]. Compared to the conventional methods for fluid-rigid coupling, the IB method is simpler and faster. As an impulse-based method, it calculates the solid boundary conditions by adding impulse forces $\mathbf{f} = (\mathbf{u}_{solid} - \mathbf{u}_{fluid})/t$ to the Euler grid to obtain the desired boundary flow conditions. This method does not need to update the fluid domain at each time step, and is able to handle the coupling between Euler (the fluid cells) and Lagrangian structures (the moving objects), which is well suited to our case. We use the IB method to simulate many dynamic phenomena as shown in our examples in Figure 11.

4.3 Material Transportation

We implemented a fluid advection method on top of the hybrid flow simulation method to describe the transportation of multiple materials in a sketched system. Our approach differs from conventional fluid advection methods (e.g., [Cohen et al. 2010; Stam 1999]) in which particles or densities are advected in one single velocity field; in our system, advection takes place in a network composed of pipes, joints, and local flow regions.

We use an n-dimensional vector $\mathbf{D}$ to describe the mixture of the n type of fluid in each pipe segment and regional cell. Each component d_i of $\mathbf{D}$ represents the volume ratio of fluid i, and the sum of all components in one vector equals one. This representation is similar to the volume of fluid method [Kang et al. 2010] used in multiple fluid simulation. $\mathbf{D}$ values are advected in the system on both the global network and local regions. The volume ratio $\mathbf{D}$ in the sequence of nodes, pipes, and regions is updated at each timestep. Network advection is based on traversing the hydraulic graph and averaging the $\mathbf{D}$ of each joint node by the flow rates of its inflow pipes. Regional advection is based on the semi-Lagrange method [Stam 1999]. Boundary cells are used to couple the flows between pipes and regions. For an inflow pipe, the volume ratio is set on the corresponding cells as boundary conditions for the semi-Lagrange advection; for an outflow pipe, the average volume ratio of its boundary cells is set as the pipe volume ratio for network advection. Each type of fluid has a color C_i to enable visualization, and fluid distributions are illustrated by blending a vertex color according to its volume ratio.

We also used traced particles for dynamic flow animations and streamlines for static keyframe illustrations. For dynamic animation, particles flow into the system from inflow nodes and flow out of the system from outflow nodes. We sort particles entering joint nodes with probabilities proportional to nodal outflows. When a particle enters a regional node, it is advected using the local velocity field. Streamlines for keyframe illustration can be automatically generated or manually seeded to produce static flow pictures.

5 Results

We demonstrated the effectiveness of the proposed system by applying it to clarify various fluid systems. The sketching interface and fluid simulation algorithms were implemented in C++ and rendered with OpenGL. We defined an 8-layer Euler grid with a resolution of 140×100 on each layer for background simulation. All illustrations ran on a PC with an 8-core 2.8 GHz CPU and 4 GB of memory. In real-time illustration, the system can reach the speed of approximately 30 frames per second (including interaction, simulation and rendering).

5.1 Heart Defects and Surgical Procedures

Our motivating application is to illustrate congenital heart diseases and the operation procedures used to treat them. Our system is particularly useful for this application because each heart has a very unique configuration. A few standard illustrations do not work; a doctor must draw a new illustration for each patient to describe the disease and operation procedure to the parents of the patient (an infant), which is very tedious and time consuming. As shown in the user study, our system can facilitate faster and more effective communication between a doctor and parents, and among medical professionals.

A congenital heart defect is a defect in the structure of the heart atria, ventricles, or vessels, which either obstructs blood flow in the heart or the vessels near it, or causes blood to flow through the heart in an abnormal pattern. As shown in Figure 9, we used our system to explain a normal heart configuration and five typical heart defects interactively. Users can use simple sketch operations to show these defects, and the abnormal blood mixtures and flow directions can be automatically generated. Each illustration can be drawn in one or two minutes. The literal explanations of these heart defects can easily be found on a heart surgery website[1] or in a medical textbook.

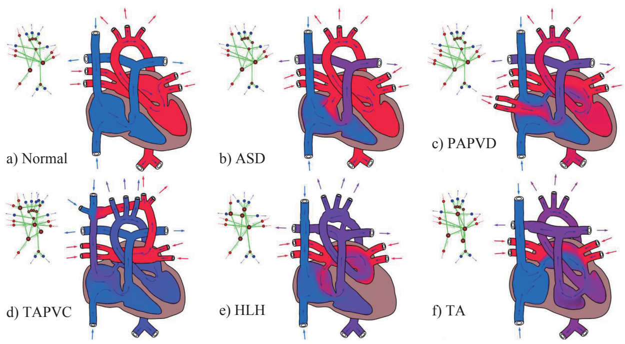

Figure 9: *Illustration of heart defects: a) normal heart, b) atrial septal defect (ASD), c) partial anomalous pulmonary venous drainage (PAPVD), d) total anomalous pulmonary venous connection (TAPVC), e) hypoplastic left heart syndrome (HLHS), and f) tricuspid atresia (TA).*

Repairing congenital heart defects such as HLHS (Figure 8 (e)) and TA (Figure 8(f)) requires surgery involving a series of complicated operations. We used our system to illustrate the surgical procedure to correct TA as shown in Figure 1. The goal is to correctly send blue venous blood to the lungs and send the red arterial blood to the body. The procedure contains three stages: Blalock-Taussig shunt, bidirectional Glenn, and Fontan procedures. The user builds a shunt, cuts and sews the superior vena cava to the right pulmonary artery, blocks the ductus arteriosus, severs the pulmonary artery from the right ventricle and sews it to the right atrium, closes the atrial septal defect, and blocks the shunt. The user can illustrate the sequence of such complex surgical operations using continuous edits and animations. Illustration of the entire sequence of operations can be completed in two to three minutes.

5.2 Physiological Systems

We used our system to illustrate various other physiological fluid systems. Figure 10a illustrates blood circulation in the kidney by

[1] http://embryology.med.unsw.edu.au/Notes/heart2.htm

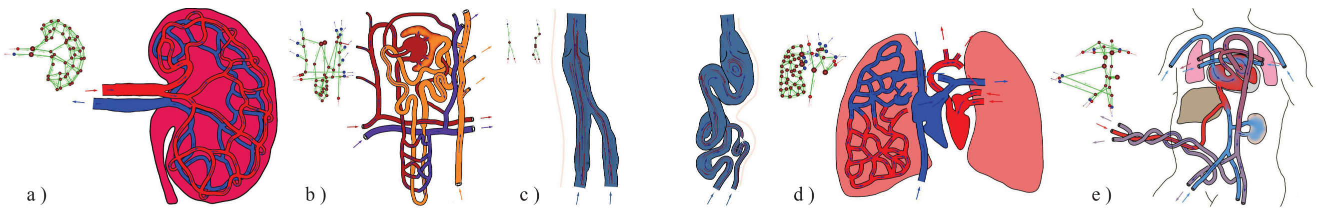

Figure 10: *Fluid systems in physiology: a) blood circulation in kidney, b) material transportation in nephron, c) varicose veins, d) oxygen transportation in lungs, and e) fetal circulation.*

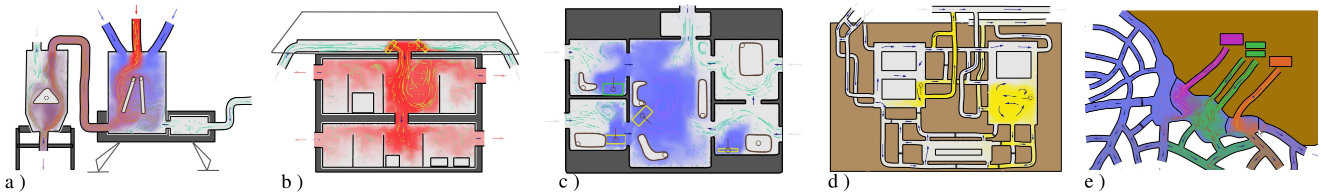

Figure 11: *Fluid systems in engineering: a) humidified agitator, b) central heating system of a house, c) indoor air temperature distribution with multiple air conditioners, d) the emergency measures to prevent poisonous gas diffusion in galleries of a coal mine, and e) pollution control in river network.*

simulating and visualizing blood flow in a vessel network. Figure 10b depicts the material transportation process in a nephron, which is the basic structural and functional element of the kidney. The nephron filters the water and soluble materials in blood that pass through it, reabsorbs what is needed, and excretes the rest as urine. Figure 10c illustrates the abnormal blood flow in a varicose vein. Comparisons between the normal vein (left) and the varicose vein (right) are given in one illustration. In a normal vein, leaflet valves prevent blood from flowing backwards, whereas in a varicose vein, valves no longer function and allow backward blood flow to enlarge the vessel. Figure 10d illustrates the oxygen transportation process in lungs and the heart. Figure 10e illustrates the complex fluid system of fetal circulation: oxygen-rich blood travels from the placenta to the fetus' body, flows through the liver, and mixes with oxygen-poor blood in the heart. The mixed blood then flows to some part of the fetus' body and finally re-enters the placenta.

5.3 Engineering Examples

We also use our system to illustrate many fluid systems in engineering. Figure 11a shows the dynamic work process of a humidified agitator. We add dynamic objects with different movement patterns to represent the paddles inside the agitator. Figure 11b shows the air circulation system of a typical two-story house and the work process of the central heater. Figure 11c shows the temperature distribution inside an apartment with multiple air conditioners in different rooms. Users can change the direction and location of the air conditioners to obtain a better cooling effect. Figure 11d shows a example of poisonous gas horizontally diffusing in the galleries of a coal mine and the corresponding emergency measures of successively closing the air doors nearby. Figure 11e is an example for environmental engineering. We show a river network with three different types of pollution produced from different chemical factories. We then show the process of pollution control by blocking the river branches and creating dams in the river delta area.

5.4 User Feedback

We recruited four expert pediatric cardiologists (30-50 years old, all male) and asked them to depict the operation procedure for a congenital heart disease condition using the system. Each doctor spent roughly 60 min for the study, 10 min for the initial orientation and questionnaire, 20 min for an explanation of and practice with the system, 15 min for the main task, and 15 min for the follow-up interview. The main task was to explain the operation procedure (1)shown in Figure 1, in which the connectivity of blood vessels is changed in a specific order. Because it was difficult to give extensive training to busy doctors, we chose to provide a step-by-step guide to them using a computer and asked them to operate on the system using another computer closely following our guide. Three doctors showed strong interest and played with the system for some time afterwards.

All of the doctors successfully completed the task following our guide, and confirmed that our system would be a useful communication tool. They reported that the user interface seemed simple enough for them to use easily after sufficient practice and training. They appreciated that they could modify the heart configuration by simple drawing and dragging, and the blood flow changed immediately afterwards. They confirmed that this process was more efficient than traditional pen-and-paper sketching in situations where they had to change blood vessel connectivity frequently, and was more effective for explaining complicated procedures to patients and nurses. The doctors saw that our system was particularly useful for explaining complicated operations such as the Fontan procedure and blood flow changes from fetal to adult circulation patterns in childbirth.

The doctors also provided various suggestions for future development. All emphasized the need to prepare many template diagrams. One expressed a strong desire to make the system completely 3D so that he could turn the heart model around to show the back. One suggested providing quick access to standard compound operations such as switching the aorta and pulmonary artery at once. Other comments were mostly related to minor implementation issues such as imperfect undo and the inability to insert text, but none questioned the adequacy of the simulation result. One wanted to specify the flow amount and direction in a blood vessel, which we plan to address in the future.

6 Limitations and Future Work

The simulations are designed to provide fast plausible enhancement for illustration purposes and are not designed for accurate simulation. Our 2D model is a simplification of the real 3D organs and vessels and it cannot model flow details in real 3D structures accurately. Therefore, our method is useful for explaining (visualizing) what a user (doctor) already knows, but will not provide additional data. The hydraulic network simulation is driven by boundary condition (external inflows) only, it remains as our future work to improve the hydraulic model to drive a flow with a force in a region or pipe. The animation only supports dynamic objects within a region, but does not support motion of a region boundary, which is necessary for illustrating many biological phenomena such as a bulging and shrinking heart. The illustrations only consider the aspect of fluid itself, and cannot represent many fluid phenomena that lie beyond the scope of fluid dynamics such as chemical reaction energies.

References

ALMEDER, C. 1999. *Hydrodynamic Modelling and Simulation of the Human Arterial Blood Flow*. PhD thesis, Vienna University of Technology.

ANGELIDIS, A., NEYRET, F., SINGH, K., AND NOWROUZEZAHRAI, D. 2006. A controllable, fast and stable basis for vortex based smoke simulation. In *Proceedings of SCA '06*, 25–32.

BRIDSON, R. 2008. *Fluid Simulation for Computer Graphics*. A K Peters.

CABRAL, B., AND LEEDOM, L. C. 1993. Imaging vector fields using line integral convolution. In *Proceedings of SIGGRAPH '93*, 263–270.

COHEN, J. M., TARIQ, S., AND GREEN, S. 2010. Interactive fluid-particle simulation using translating eulerian grids. In *Proceedings of I3D '10*, 15–22.

DAVIS, J., AGRAWALA, M., CHUANG, E., POPOVIĆ, Z., AND SALESIN, D. 2007. A sketching interface for articulated figure animation. In *ACM SIGGRAPH 2007 courses*.

DAVIS, R. C., COLWELL, B., AND LANDAY, J. A. 2008. K-sketch: a 'kinetic' sketch pad for novice animators. In *Proceeding of CHI '08*, 413–422.

DAVIS, R. 2007. Magic paper: Sketch-understanding research. *IEEE Computer 40*, 9, 34–41.

EBERT, D. S., SOUSA, M. C., GOOCH, A., AND STREDNEY, D. 2005. Computer-generated medical, technical, and scientific illustration. In *ACM SIGGRAPH 2005 Courses*.

ERNERFELDT, E., 2008. Phun. Website: http://www.phunland.com/.

FORMAGGIA, L., AND VENEZIANI, A. 2003. Reduced and multiscale models for the human cardiovascular system. *Lecture notes VKI, Lecture Series 2003-07 MOX Report 21*.

GINGOLD, Y., IGARASHI, T., AND ZORIN, D. 2009. Structured annotations for 2d-to-3d modeling. *ACM Trans. Graph. 28*, 5, 148:1–148:9.

IGARASHI, T., MATSUOKA, S., AND TANAKA, H. 1999. Teddy: A sketching interface for 3d freeform design. In *Proceedings of SIGGRAPH '99*, 409–416.

IGARASHI, T., MOSCOVICH, T., AND HUGHES, J. F. 2005. As-rigid-as-possible shape manipulation. *ACM Trans. Graph. 24*, 3, 1134–1141.

IJIRI, T., OWADA, S., AND IGARASHI, T. 2006. Seamless integration of initial sketching and subsequent detail editing in flower modeling. In *Computer Graphics Forum*, vol. 25, 617–624.

IRVING, G., GUENDELMAN, E., LOSASSO, F., AND FEDKIW, R. 2006. Efficient simulation of large bodies of water by coupling two and three dimensional techniques. *ACM Trans. Graph. 25*, 3, 805–811.

KANG, N., PARK, J., NOH, J., AND SHIN, S. Y. 2010. A hybrid approach to multiple fluid simulation using volume fractions. *Computer Graphics Forum 29*, 2, 685–694.

LEE, S.-H., SIFAKIS, E., AND TERZOPOULOS, D. 2009. Comprehensive biomechanical modeling and simulation of the upper body. *ACM Trans. Graph. 28*, 4, 99:1–99:17.

MCCANN, J., AND POLLARD, N. 2009. Local layering. *ACM Trans. Graph. 28*, 3, 84:1–84:7.

MCLOUGLIN, T., LARAMEE, R. S., PEIKERT, R., POST, F. H., AND CHEN, M. 2010. Over two decades of integration-based geometric flow visualization. *Computer Graphics Forum 29*, 6, 1807–1829.

MITRA, N. J., YANG, Y.-L., YAN, D.-M., LI, W., AND AGRAWALA, M. 2010. Illustrating how mechanical assemblies work. *ACM Trans. Graph. 29*, 4, 58:1–58:12.

MITTAL, R., AND IACCARINO, G. 2005. Immersed boundary methods. *Annual Review of Fluid Mechanics 37*, 1, 239–261.

MÜLLER, M., CHARYPAR, D., AND GROSS, M. 2003. Particle-based fluid simulation for interactive applications. In *Proceedings of SCA '03*, 154–159.

NOBILE, F. 2009. Coupling strategies for the numerical simulation of blood flow in deformable arteries by 3d and 1d models. *Mathematical and Computer Modelling 11-12*, 2152–2160.

OKABE, M., ANJYO, K., IGARASHI, T., AND SEIDEL, H.-P. 2009. Animating pictures of fluid using video examples. *Computer Graphics Forum 28*, 2, 677–686.

PROMETECH, 2008. Physicafe. http://www.prometech.co.jp/english/products/physicafe.html.

PURHO, P., 2008. Crayon physics deluxe. http://www.crayonphysics.com/.

SCHROEDER, D., COFFEY, D., AND D.KEEFE. 2010. Drawing with the flow: A sketch-based interface for illustrative visualization of 2d vector fields. In *Proceedings of SBIM '2010*, 49–56.

SEWALL, J., WILKIE, D., MERRELL, P., AND LIN, M. C. 2010. Continuum traffic simulation. *Computer Graphics Forum 29*, 2, 439–448.

STAM, J. 1999. Stable fluids. In *Proceedings of SIGGRAPH '99*, 121–128.

TREUILLE, A., LEWIS, A., AND POPOVIĆ, Z. 2006. Model reduction for real-time fluids. *ACM Trans. Graph. 25*, 3, 826–834.

VAINIO, T., HAKKARAINEN, K., AND LEVONEN, J. 2005. Visualizing complex medical phenomena for medical students. In *CHI '05 extended abstracts*, 1857–1860.

VAN WIJK, J. J. 2002. Image based flow visualization. *ACM Trans. Graph. 21*, 3, 745–754.

YU, Q., NEYRET, F., BRUNETON, E., AND HOLZSCHUCH, N. 2009. Scalable real-time animation of rivers. *Computer Graphics Forum 28*, 2, 239–248.

ZELEZNIK, R. C., HERNDON, K. P., AND HUGHES, J. F. 1996. Sketch: an interface for sketching 3d scenes. In *Proceedings of SIGGRAPH '96*, 163–170.

ZHANG, E., MISCHAIKOW, K., AND TURK, G. 2006. Vector field design on surfaces. *ACM Trans. Graph. 25*, 4, 1294–1326.

ZÖCKLER, M., STALLING, D., AND HEGE, H. C. 1996. Interactive visualization of 3d-vector fields using illuminated stream lines. In *Proceedings of VIS '96*, 107.

Appendix

This section briefly describes how to construct and solve the linear system of the flow network based on [Almeder 1999]. For a given hydraulic graph $G(n_n, n_e)$, vector Q_e represents the flow rate of n_e pipes, and vector Q_n describes the external sources in n_n nodes. This vector can be specified based on user operations and is used as the known values on the right side of the linear equation systems. For each outflow node, a virtual edge must be constructed linking the node and the collecting node as in [Almeder 1999]. Isolated nodes are eliminated from the network calculation.

Velocities and pressures in the hydraulic network are solved based on the following laws:

- At each node, the sum of the flow rate of the outgoing pipes equals the sum of the flow rate of the incoming pipes and the external inflows.
- There is a linear relationship between the flow rate and the pressure drop in a pipe. The pressure drop is the difference in pressure between the head and tail node.

The first law states that the network is a closed system following the Kirchhoff rule, and there is no fluid loss in the network. The second law ignores fluid details such as turbulence inside the pipes and treats fluid in pipes as a laminar flow. With the node-edge matrix constructed in Sec.4.1, the two laws can be described in vector form as follows:

$$Q_n = -\mathbf{M}Q_e, \tag{3}$$

$$P_e = -\mathbf{M}^T P_n. \tag{4}$$

Eq.3 describes the balance of inflow, outflow, and external flow. Eq.4 describes the relationship between a drop in pipe pressure and node pressure, in which P_n represents the pressure of each node and P_e represents the drop in pressure in each pipe.

The pipe laminar flow theory can be used to set up the relationship between a drop in pressure and flow rate in a pipe. When fluid flows through a pipe, drop in pressure is caused by friction between the fluid and the pipe walls and by fluid viscosity. It can be expressed as:

$$p_{drop} = \frac{\rho l v^2 \lambda}{2d}, \tag{5}$$

in which ρ, l, v, λ and d are fluid density, pipe length, mean flow velocity, wall friction coefficient, and pipe diameter, respectively. Flow velocity can be calculated as:

$$v = \frac{Q}{S} = \frac{4Q}{d^2\pi}, \tag{6}$$

and the wall friction coefficient can be calculated based on the Reynolds number:

$$\lambda = \frac{64}{Re} = \frac{64\nu}{vd}. \tag{7}$$

By substituting Eq.6 and Equation 7 into Equation 5, the relationship between pressure and flow can be expressed as:

$$p_{drop} = \frac{128\rho\nu l Q}{\pi d^4}. \tag{8}$$

To wrap Equation 8 into a matrix form, we can use the following:

$$Q_e = \mathbf{D}_e P_e, \tag{9}$$

in which $\mathbf{D}_e$ is a diagonal matrix with the diagonal element $D_{ii} = Q_e^i / p_{drop}^i = \pi d^4 / 128\rho\lambda l$.

Based on Equation 3, 4 and 9, a linear equation system with unknown p_n can be expressed as:

$$\mathbf{M}\mathbf{D}_e\mathbf{M}^T P_n = Q_n. \tag{10}$$

This linear equation system can be solved using the Gauss-Seidel iteration method. With the solved pressure on each node, the drop in pipe pressure can be calculated based on Equation 4.Next, the flow rate and velocity can be calculated using Equation 9 and Equation 6. The solved flow velocities in pipes can then be used as the boundary conditions of the regional solvers and to help to dynamic visualization of flows in the entire system.