

# Real-time Synthesis of Bleeding for Virtual Hysteroscopy

János Zátanyi<sup>1</sup>, Rupert Paget<sup>1</sup>, Gábor Székely<sup>1</sup>, and Michael Bajka<sup>2</sup>

<sup>1</sup> Swiss Federal Institute of Technology Zürich, CH-8092 Zürich, Switzerland  
{zatonyi, rpaget, szekely}@vision.ee.ethz.ch

<sup>2</sup> Clinic of Gynecology, Dept. OB/GYN, University Hospital of Zürich, Switzerland

**Abstract.** Simulating bleeding in a virtual reality surgical simulator is an important task that still has not found a visually appealing solution. Bleeding in a simulator not only tests a surgeon to deal with critical issues, but also affects the environment by obscuring the view in which the surgeon has to operate. For any virtual reality surgical simulator, bleeding has to be treated, while at the same time the bleeding has to be responsive to any feedback that the surgeon may be conducting to the virtual reality environment. And all this has to be performed in real-time, i.e. at frame-rate. In this paper we present a methodology for solving this particular problem and show preliminary results of real-time visualization of bleeding in a dynamic virtual reality environment.

## 1 Introduction

Hysteroscopy is the standard visualization of the inner surface of the uterus performed by using one hull containing both the endoscope and the surgical instrument through the cervix into the uterus and applying fluid under pressure for establishing the hydrometra. It is the second most often performed endoscopic procedure after laparoscopy in gynecology [1]. Diagnostic hysteroscopy is used for clarification of pathologic alterations in case of symptoms or findings by imaging modalities like irregular bleeding, endometrial thickening, suspected tumors and problems of infertility or sterility. If necessary, it may be subsequently continued with therapeutic hysteroscopy, e.g. for removal of polyps and myomas, endometrial ablation, resection of adhesions or uterine septa, catheterism of the fallopian tube or removal of intrauterine device [2].

The side effects and iatrogenic injuries depend on the procedure that is performed: while the pure diagnostic hysteroscopy presents almost no major problems, therapeutic hysteroscopy is associated with a small number of well-known serious complications. They primarily result from either the inadequate technique of the surgeon, or the fluid overload of the patient. The only way to acquire sufficient experience is with the repetitive training of specific skills, procedures and complication management. However, patient involvement should be avoided during early phase of the surgeon's learning curve.

Currently the basic visio-spatial and manipulative skills are taught by in vitro methods using inanimate objects such as sheep bladders or bell peppers [3].

These units allow the surgeon to learn how to navigate under monoscopic visual feedback, as well as perform basic manipulative components of an intervention. In this way the surgeon develops competence in completing a particular task, but because the real-life effect is lost, one obtains only a limited training in dexterity and surgical problem solving. Primarily, there is a lack of realistic tissue reactivity and one cannot experience the complexities of abnormal anatomy or pathologic situations.

It is believed, that a reasonably realistic virtual-reality-based endoscopic simulator training for therapeutic hysteroscopies could contribute to a reduced rate of complications. The ultimate advantage of such a simulator is the potential to provide a realistic and configurable training environment that bridges the gap between basic training and performing the actual intervention on patients, without any restriction on repetitive training. However, the simulator systems proposed to date do not achieve the necessary level of realism that is required for this technology to be widely accepted in the medical community [3,4].

In the simulator, the anatomy and some well defined pathologies must be represented. Realistic real-time simulation of the changes in the operational site due to surgical actions and photo-realistic rendering must be achieved, including the control of the hydrometra by manipulating the liquid inflow and outflow. Finally, the manipulator should allow realistic tactile sensations (realized by force-feedback). In contrast to other virtual reality applications, creating the appropriate virtual environment is straightforward. The simulator with its monitor and manipulator corresponds directly to the setup in the true surgical situation.

In a hysteroscopic simulator, special interest is in the simulation of diffuse intra-uterine bleeding, obscuring the view of the surgeon, until the correct actions (adjusting the inflow and outflow of liquid) are performed. It has to be emphasized, that the mechanism of fluid management requires specific skills from the surgeon. On one side, proper fluid management is crucial to the patient's safety, on the other side, it is the only way to keep the operation site visible. Consequently, we face the task of synthesizing bleeding, which can only be made realistic if we are also able to model its specific dynamic environment, in which an almost continuous current of the distension liquid is kept alive.

## 2 Methodology

The realistic synthesis of bleeding is an important problem without a defined solution. Temporal texture synthesis [5–10] is one possible approach to solve this problem. However, these techniques are limited in their ability to mimic certain temporal phenomena, and do not have the full capability of synthesizing a spatially stochastic, temporally heterogeneous texture like bleeding. Therefore we have chosen a more promising fluid dynamics approach [11]. This choice is motivated by the fact that in the case of hysteroscopy, the bleeding takes place in the cavum uteri distended by liquid.

The precise mathematical equations describing the behavior of fluid flows are the so-called Navier-Stokes equations [12]. They are non-linear partial differen-

tial equations and an analytical solution to them is feasible only for the very simple cases. With the rapid evolution of computers, numerical solutions to the equations have come to the front, establishing the domain of computational fluid dynamics (CFD). This branch of fluid dynamics gives a cost-effective way for accurate, real flow simulations specific for engineering purposes. Alternatively, fluid solvers from computer graphics domain provide the possibility to achieve fluid-like effects in real-time, where precise physical accuracy is not as important as just plain visual fidelity.

Foster and Metaxas [13] used a coarse grid on which they invoked a simplified discretized version of the Navier-Stokes equations. The non-linear partial differential equations are solved using an explicit finite differencing algorithm. However this induces instabilities if the time-step is chosen too high. This limitation also means that the speed of the simulation is also restricted. Stam [14] proposed an unconditionally stable fluid solver. More specifically, instead of the explicit Eulerian integration scheme both semi-Lagrangian [15] and implicit methods are used to solve the Navier-Stokes equations. This provided the basis on which to build real-time interactive visualization of realistic fluid flows.

## 2.1 Introducing the Basic Model

The state of a volume of fluid is determined by its density, temperature and velocity. In this basic model both the density and the temperature of the fluid are considered constant. The behavior of the velocity field is mathematically expressed by the Navier-Stokes equations:

$$\nabla \cdot \mathbf{u} = \mathbf{0} \quad , \quad (1)$$

$$\frac{\partial \mathbf{u}}{\partial t} = -(\mathbf{u} \cdot \nabla) \mathbf{u} + \nu \nabla^2 \mathbf{u} + \mathbf{f} \quad , \quad (2)$$

where  $\mathbf{u}$  is the velocity vector field,  $\nu$  represents the kinematic viscosity coefficient and  $\mathbf{f}$  corresponds to the vector field of user-induced forces.

This is the compact form of the Navier-Stokes equations. The quantities  $\mathbf{u}$  and  $\mathbf{f}$  are varying both in space and time. The spatial coordinates can be considered either as 2D  $(x, y)$  or 3D  $(x, y, z)$  coordinates. Equation (1) means that the fluid should conserve mass, while in (2) the momentum conservation is formulated. For the derivation of the equations we refer the reader to [12].

The velocity field is used to move around a substance within the fluid, whereby the evolution of the substance is governed by the following transport equation:

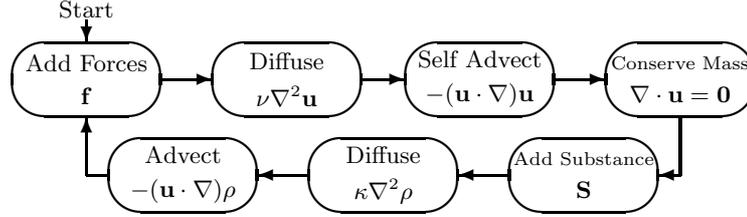
$$\frac{\partial \rho}{\partial t} = -(\mathbf{u} \cdot \nabla) \rho + \kappa \nabla^2 \rho + \mathbf{S} \quad , \quad (3)$$

where  $\rho$  is the scalar field of the density of the substance,  $\kappa$  is the diffusion constant and  $\mathbf{S}$  is the scalar field of the amount of substance injected. With this model both the motion of the fluid and the related propagation of substances (like blood) can be handled.

## 2.2 Method of Solution

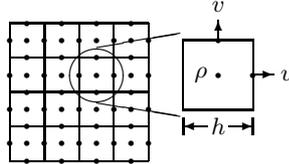
There are various ways of implementing the above model [16–18]. The differences are mainly in regard to possible improvements in either the visual appearance or computational efficiency. In our implementation we optimize the methodology for real-time visualization of bleeding in the uterine cavity.

The general method is to solve all the terms in (2) and (3) sequentially. This loop is illustrated schematically in Fig. 1.



**Fig. 1.** Flowchart of the solution. Top line contains the evolution of the velocity field, Eqns. (1) and (2). Bottom line contains the evolution of the substance field, Eqn. (3).

In practice, each quantity is defined on a spatially discretized domain. In the first stage we implemented the model in a 2D environment, and our domain is a squared grid. Instead of the uniform grid configuration considered in [14, 18] we used the so-called staggered grid arrangement as in [13, 16]. It means – as shown in Fig. 2 – that the density of the substance and the external forces are defined at the center of each cell, while the velocity field is orthogonally separated into its scalar components and defined on the faces of the grid cells. In accordance with the results in [16], our experiments also showed the visual benefits of less numerical dissipation by applying this staggered arrangement.



**Fig. 2.** Spatially discretized staggered grid configuration

The simulation is advanced by updating the grid from a previous status over a predefined time step  $\Delta t$ . First, the velocity components of the fluid are updated in four major steps. Step one is the addition of the external force fields to the velocity field. The external force fields can contain e.g. pre-configured forces or more specifically in our case the forces induced by manipulating the liquid inflow and outflow during the surgical intervention. The forces are multiplied by the time-step, then averaged to the grid faces on which the velocities are defined and the corresponding scalar components are added to the velocities.

The second step is the diffusion of the velocity field. It is solved in the same way as for the diffusion of the substance in step two of the density solver. However instead of a diffusion rate  $\kappa$ , a viscosity  $\nu$  is used, and each component of the velocity field is solved separately. For easier conceivability the method is explained later as part of the density solver.

The third step is the self-advection. It can be interpreted as the velocity field moving itself. It is a non-linear problem, but Stam [14] showed how a stable solution, viable for computer graphics, can be obtained using the semi-Lagrangian integration scheme [15]. Basically each velocity component is traced back in time through the velocity field to find where it came from. This can be done using either a linear particle tracer or a second order Runge-Kutta method [19]. From our experiments the more elaborate latter one did not provide the improvements worthy of the increase in computation time. At the past points, interpolation is needed to obtain the past velocities. This may be achieved via simple linear, or the so-called monotonic cubic interpolation method as proposed in [16]. Although the monotonic cubic interpolation scheme reduces the simulation speed considerably, the realism of the appearance of the fluid improved adequately (especially when used during the density solution step). Finally the calculated velocities are transported back to where the back-tracing originated from.

The fourth step is mass conservation according to (1). It is a physical constraint that every fluid has to conserve mass, meaning that the flow into a cell should be equal to the flow coming out of the cell. After the preceding steps (adding external forces, viscous diffusion, self-advection) it is not fulfilled. Mass conservation is important also from the computer graphics point of view, as it forces the flow to swirl and have vortices enhancing the flow's realism.

In order to obtain mass conserved velocity field with appealing vortices, the mathematical theorem known as Helmholtz-Hodge decomposition is used [12]. The application of the theorem incorporates the task of solving a sparse linear equation system. There are diverse methods for computing a solution [14], but we found that the simple iterative solver, called Gauss-Seidel relaxation [18], was the fastest while still giving visually satisfactory results. To minimize numerical dissipation and maintain the realism of the flow, we also added vorticity confinement [16] which enhanced these vortices through an additional force component.

Let us now move ahead to the second major part of the model. The density equation (3) governs the evolution of the substance injected into the fluid. The method of solution is similar to the one introduced previously. We start with an initial density field, which can be empty, and solve the terms in (3) sequentially. The first step is the addition of the density values of an external source, which are multiplied by the time-step and added to the density field. In our case, external source is e.g. the spurting blood from vessel on the wall of the uterus.

The second step is to solve for the diffusion of the density. Due to the discretized domain, the second spatial derivative is approximated by finite differences. The straightforward implementation of a diffusion task can cause stability problems during the simulation if the diffusion rate or the time-step are set too large or the grid spacing is too small. To overcome this problem Stam [14] pro-

posed an implicit, thus stable solution method for the diffusion term: find the density field which would diffuse backward in time to give the densities we started with. In this way we again face the problem of solving a linear equation system. Due to the reasons mentioned above the Gauss-Seidel relaxation [18] was used here as well, but here 4-5 iterations are enough for a realistic impression.

There is one more term in (3) which has to be solved. It states that the density of the substance should follow and be advected by the – previously updated – velocity field. In order to preserve the stability of the solver, here again the semi-Lagrangian integration scheme is used [14]. We trace back the midpoints of each cell through the velocity field. Then the new density values are interpolated at these points and their values are transferred to the grid centers where we started at. For our purposes we used the monotonic cubic interpolation scheme [16] due to its advantageous properties in giving a the realistic look to the substance.

### 3 Results

Figure 3 gives a visual comparison between real bleeding during hysteroscopic surgery when fluid inflow and outflow are stopped and our simulated version. The synthesis has high visual fidelity, but what is more important, the synthesis is being performed in real-time (24 frames/sec) on a  $55 \times 55$  grid, and is responding to changes in the virtual environment.

One aspect that is essential in hysteroscopy simulation is the accounting for the effect of inflow and outflow fluid management. In our 2D representation, we account for this flow by inducing constant velocity into the grid at the lower left and right corners. For outflow, we allow the velocities to cross the boundary in the middle of the lower boundary. In figure 4 we show what happens to the blood when both flows are switched on in frame 2. Although this does not yet give a fully realistic representation of what happens in real hysteroscopic surgery, it does already demonstrate that the simulated blood can be affected by its dynamic virtual environment in real-time. A realistic representation of the effect of fluid management on the bleeding will be achieved when the simulation is performed in a 3D virtual environment representative of what is required for hysteroscopic surgery.

Current limitation of the model is that it is not yet sufficiently adapted to an inhomogeneous fluid environment capable to capture the visual effects originating from the interaction between blood and distension fluid.

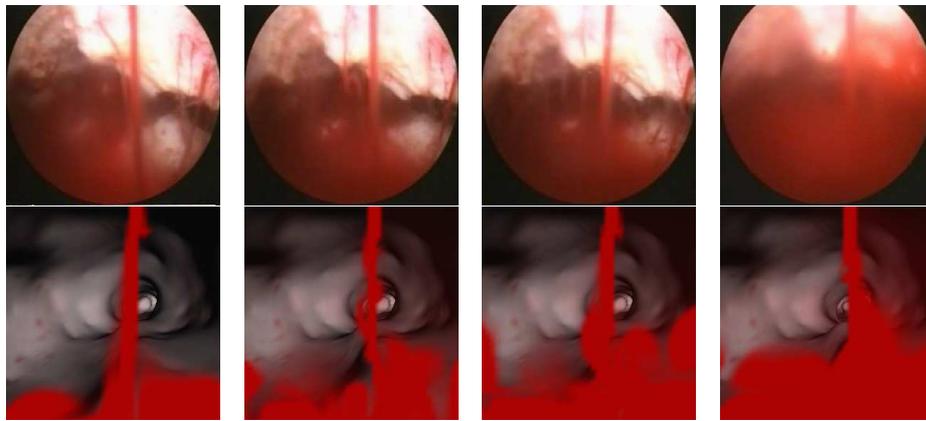
### 4 Conclusion and Future Research

We have explored the current research in synthesizing temporal phenomena in computer graphics. From this research we have presented a methodology that has been optimized for synthesizing bleeding in a virtual reality hysteroscopy surgical simulator. Preliminary results show that the presented technique, based on the Navier-Stokes fluid equations, can be solved in real-time (i.e. frame-rate or 24 frames/sec). This fluid technique also allows the bleeding to be interactive and

responsive to any changes made to the virtual reality environment in real-time. Future work will focus on presenting further enhances particular to synthesizing various aspects of bleeding, and explicitly integrating the inflow and outflow of the fluid induced by the surgeon through the simulator.

## Acknowledgment

This research has been supported by the NCCR CO-ME of the Swiss National Science Foundation.



**Fig. 3.** Top row: original images from hysteroscopy surgery during bleeding when fluid inflow and outflow are stopped. Bottom row: frames from real-time synthesis.<sup>1</sup>



**Fig. 4.** Interacting with the environment: simulating inflow and outflow.<sup>1</sup>

---

<sup>1</sup> Color version of the figures can be found at <http://www.vision.ee.ethz.ch/>

## References

1. Sierra, R., Székely, G., Bajka, M.: Generation of pathologies for surgical training simulators. In: *Procs. of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention*. Volume 2., Springer Verlag (2002) 202–210
2. Köchli, O.R., ed.: *Hysteroscopy : state of the art*. Volume 20 of *Contributions to gynecology and obstetrics*. Basel : Karger (2000)
3. Montgomery, K., Bruyns, C., Wildermuth, S., Heinrichs, L., Hasser, C., Ozenne, S., Bailey, D.: Surgical simulator for operative hysteroscopy. *IEEE Visualization 2001* (2001) 14–17
4. Müller-Wittig, W.K., Bockholt, U., Arcos, J.L.L., Oppelt, P., Stähler, J., Voss, G.: Lahystotrain - VR-based intelligent training environment for laparoscopy and hysteroscopy. In: *Virtual Reality International Conference (VRIC)*. Volume 3. (2001) 225–233
5. Bar-Joseph, Z., El-Yaniv, R., Lischinski, D., Werman, M.: Texture mixing and texture movie synthesis using statistical learning. *IEEE Transactions on Visualization and Computer Graphics* **7** (2001) 120–135
6. Doretto, G., Soatto, S.: Editable dynamic textures. In: *ACM SIGGRAPH 2002 Sketches and Applications*, San Antonio, Texas (2002)
7. Oppenheimer, P., Gupta, A., Weghorst, S., Sweet, R., Porter, J.: The representation of blood flow in endourologic surgical simulations. In: *Proceedings of Medicine Meets Virtual Reality*. (2001) 365–371
8. Perlin, K., Neyret, F.: Flow noise. *Siggraph Technical Sketches and Applications* (2001) 187
9. Schödl, A., Essa, I.: Controlled animation of video sprites. In: *First ACM Symposium on Computer Animation in Conjunction with ACM SIGGRAPH 2002*, San Antonio, TX, USA (2002)
10. Szummer, M.: Temporal texture modeling. Master's thesis, MIT Media Lab Perceptual Computing (1995)
11. Raghupathi, L.: Simulation of bleeding and other visual effects for virtual laparoscopic surgery. Master's thesis, University of Texas at Arlington (2002)
12. Chorin, A., Marsden, J.E., eds.: *A Mathematical Introduction to Fluid Mechanics*. 2nd edn. Springer-Verlag, New York (1990)
13. Foster, N., Metaxas, D.: Modeling the motion of a hot, turbulent gas. In: *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co. (1997) 181–188
14. Stam, J.: Stable fluids. In: *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, ACM Press/Addison-Wesley Publishing Co. (1999) 121–128
15. Staniforth, A., Cote, J.: Semi-Lagrangian integration schemes for atmospheric models: A review. In: *Monthly Weather Review*. Volume 119. (1991) 2206–2223
16. Fedkiw, R., Stam, J., Jensen, H.W.: Visual simulation of smoke. In: *Procs. of ACM SIGGRAPH 2001*, ACM Press (2001) 15–22
17. Stam, J.: A simple fluid solver based on the FFT. *Journal of Graphics Tools* **6** (2001) 43–52
18. Stam, J.: Real-time fluid dynamics for games. In: *Proceedings of the Game Developer Conference*. (2003)
19. Press, W.H., Flannery, B.P., Teukolsky, S., Vetterling, W.T., eds.: *Numerical Recipes in C. The art of scientific computing*. Cambridge University Press, Cambridge (1988)