HAVE 2007 – IEEE International Workshop on
Haptic Audio Visual Environments and their Applications
Ottawa – Canada, 12-14 October 2007

Real-Time Fluid Interaction with a Haptic Device

Javier Mora[1], Won-Sook Lee[2]
[1]SITE, University of Ottawa, jmora091@uottawa.ca
[2]SITE, University of Ottawa, wslee@site.uottawa.ca

Abstract – **Imagine an interactive videogame in which the player impersonates the role of a witch that desires to create a potion. Through a haptic probe, shaped as a baton, users are able to stir and feel the magical fluid inside a bowl. As players follow the potion recipe, they are able to feel how the fluid changes its viscosity, density, velocity and other properties. This haptic interface enables users to interact with the digital world and receive realistic kinesthetic and tactile cues in a computer-generated environment. The novel techniques described on this paper may bring this simulation to a higher degree of human-computer interaction and immersion. We present an experimental framework for assessing haptic effects in stirred fluid simulations. The system imitates the physical forces generated by the real-time fluid animation, stirring movements and fluid changes. We discuss the integration of both the haptics and graphics workspaces for an efficient interaction.**

Keywords – Haptic Feedback, Fluid Simulation, Gesture Recognition, Gaming.

I. INTRODUCTION

Fluid animation is of great popularity in computer graphics and animation. However, it is difficult to achieve a real-time stable simulation due to the heavy computation required to solve the non-linear Navier-Stokes equation, and therefore interactive fluid animation has been problematic so far. Our motivation is to produce a system that brings human-computer interaction to real-time fluid animations, so that users can appreciate and feel the properties of a fluid simulation via a haptic interface.

Haptics refers to the technology which stimulates the users' sense of touch. Haptics allow users to literally touch and feel characteristics about computer-generated objects such as texture, roughness, viscosity, elasticity, and many other properties. The human tactile and kinesthetic senses are stimulated through computer-controlled forces which convey to the users a sense of natural feel about a virtual or remote environment. The applications of haptic technology are widespread. For instance, in combination with audio and video displays, haptics technology may be used to train people for tasks requiring hand-eye coordination, such as surgery or ship docking maneuvers. The videogame industry can also serve from this technology. Nintendo's recent Wii games [26] are an example of the industry's interest for higher interactive applications. Haptics may be used to enhance the entertainment value of such videogames. Haptics would allow players to feel the physical properties of in-game objects, adding an extra level of interaction that traditional interface devices do not offer. Currently, there is a variety of haptic interfaces available. Some devices, such as the Logitech Rumblepad [20], offer 2 degrees of freedom (DOF) interactivity and display simple haptic effects to the users — open-loop vibrotactile feedback, predefined force feedback signals. More sophisticated devices, such as the Phantom Omni and Novint Falcon, offer a higher level of interactivity by providing 6 DOF input and 3 DOF feedback to the users. Our paper focuses on two main issues: real-time fluid animation and haptic interaction with the fluid. In addition, we also discuss haptic gesture recognition as an interactive application for haptic games.

In Section 2 of this paper we present a literature review on haptic games as well as on real-time fluid animations. In Section 3 we present Jos Stam's real-time fluid dynamics [1] from which we based our graphical simulations and force-feedback calculations. The integration of graphics and haptics workspaces is discussed in Section 4. In addition, the experiment results are shown in Section 5. As an application in computer game, a haptic gesture recognition module is described in Section 6. We present conclusions in section 7.

II. RELATED WORK

The amount of literature regarding haptic technology and rendering has increased substantially in recent years. A more complete background on haptic rendering and haptics in general can be found in other articles [2][3][4][5]. Most typical examples of real-time haptic applications are in games. Experimental haptic games such as HaptiCast [6], and Haptic Battle Pong [7] have been generating brainstorming ideas for assessing haptic effects in game design. In HaptiCast, players assume the role of a wizard with an arsenal of haptically-enabled wands which they may use to interact with the game world. Haptic Battle Pong uses force-feedback to haptically display contact between a ball and a paddle. However, interaction with the game environment is limited since players can feel only the transient forces generated as the paddle strikes the ball.

There has also been some other work concerning the integration of haptics into a 3D game engine. Nilsson and Aamisepp [8] explain the relevance of incorporating haptics in a 3D engine and a plug-in for Crystal Space [9] was developed to demonstrate this integration successfully. However, haptic interaction in the context of 3D gaming was not well explored by this project. Other efforts [10] to combine haptic and graphical rendering are ambitious, but do not contain features which are desirable for 3D game development. Kauffman et al. [25] present interesting haptic sculpting tools to expedite the deformation of B-spline surfaces with haptic feedback and constraints, but they do not explore any feedback integration with fluid simulations. Dobashi and his team [24] created a model that approximates real-world forces acting on a fishing rod or kayak paddle by doing part of the math in advance of the simulation: the forces associated with different water velocities and different positions for the paddle or fishing lure were pre-calculated and saved in the database. In addition, their simulation is based on a much larger setup, including two projection screens and large haptic equipment. In contrast to this, our intention is to enable to render real-time fluid calculations on a personal computer or a laptop with a low-end desktop haptic device. To cope with these resource limitations, we take an approach in simulating real-time 3D fluids, by rendering several deformable 2D layers of fluid simulation connected at different heights.

Jos Stam [1] was the first to demonstrate a Navier-Stokes fluid simulation at interactive rates by using a grid-based numerical method free from timestep restrictions [11]. This 2D-based implementation is also shown in their experiment. It is a major achievement that enables real-time fluid dynamics. We compute forces from this type of simulation and further explain it in section 3. Baxter and Lin [11] present a complete thorough section of related work on fluid-haptic topic. They demonstrate an interesting method for integrating force feedback with interactive fluid simulation [11]. They also calculate the force-feedback from the Navier-Stokes equations of fluid motion, and adapt their fluid-haptic feedback method for use in a painting application that enables artists to feel the paint based on flat surface. In our paper, we focus on the force feedback that results from the interaction with a constrained pool of fluid. Karljohan Lundin et al. [12] present a case study where new modes for haptic interaction are used to enhance the exploration of Computational Fluid Dynamics (CFD) data. However, we are more concerned about addressing a different scenario, in which the haptic interface interacts with a bounded fluid simulation that is fast enough to be used in real-time interaction applications.

We explore haptic ambient forces to represent differences in fluid densities. An ambient force is a global strength effect that surrounds the haptic probe, regardless of collision with any surface. In addition, we adapt our method to be integrated with a spring-net deformable surface, enabling users to perceive the ripples of interaction in a 3D perspective.

III. REAL-TIME FLUID SIMULATION

For computer animators, the main concern is to achieve an efficient and visually plausible effect of a stable real-time fluid interaction, while physical accuracy is of second priority [1]. In our paper, the real-time fluid simulation part is based on Jos Stam's previous work [13][14]. Like most fluid animation methods, it is based on the classic Navier-Stokes Equation. The Navier-Stokes Equation can be presented in both velocity and density fields [1]:

$$Velocity: \frac{\partial u}{\partial t} = -(u \cdot \nabla)u + \nu\nabla^2 u + f \qquad (1)$$

$$Density: \frac{\partial \rho}{\partial t} = -(u \cdot \nabla)\rho + k\nabla^2 \rho + S \qquad (2)$$

These equations describe the behavior of fluid at one point in a fluid volume. Here, u is the vector-valued velocity at that point, t is time, ν is the kinematic viscosity of the fluid, f represents the external force, ρ represents its density, S represents the external source that is being added to the fluid, p is pressure, k represents a constant at which density tends to diffuse, "$\cdot$" denotes a dot product between vectors, and "∇" denotes the vector of spatial partial derivatives. The two similar representations of Navier-Stokes equations indicate that we could solve both velocity and density fields in a similar fashion. The incompressibility property of fluids determines that there is an additional constraint, known as the continuity equation. This serves to ensure the conservation of mass. It is a constant-density fluid which could be presented as $\nabla \cdot$u=0. Due to the application purpose of this paper, we do not describe too many details about the mathematic principles behind the Navier-Stokes equations. In Jos Stam's previous work, the main process for computing density consists of three major steps: adding force, diffusing fluids, and moving fluids. At initialization, the fluids are discretized into computational grids. The velocity field is defined at the center of each cell. For each step, the fluid solver would calculate the parameters in the equation in order to make the simulation real-time. His method applies these three main steps for both the fluid's density and velocity properties due to their similarity.

The back-tracing method [1] is the main reason for making the interactive fluid stable and efficient, but it is also the main reason for causing not high-level accuracy and unrealistic visual effects. However the visual effects are still quite acceptable.

We integrate Jos Stam's 2D real-time fluid simulation method into a 3D deformable surface with depth effect and force feedback. When users touch the fluid surface, through the haptic interface, they can perceive the resulting surface deformations. When they stir the fluid, they can see changes in the fluid's density and velocity and simultaneously feel the resulting force. The force felt depends on the velocity, direction, density and viscosity properties. These values are calculated and displayed each time the graphic workspace is updated.

MULTIPLE FLUID SIMULATION

A source represents a substance with given properties that enters the base fluid simulation. Our system also allows for the combination of multiple sources on top of the base fluid. Therefore, different calculation grids are maintained for each substance. Each source has its own characteristics and colors. The resulting rendered force is a weighted combination of each source's grid involved in the mix. Figure 2 shows an initial red source which is later mixed with a denser green source. The result is a yellowish blend which combines the contributed haptic properties of both sources. In order to give the idea of fluid 3D depth, multiple layers of this simulation are rendered at different heights. An alpha channel is also used by each deformable surface layer in order to incorporate fluid translucency into the simulation. Alpha channels are dynamically adjusted based on density values during the simulation. The detailed methods for integration will be described in the following section.

IV. HAPTIC AND FLUID INTEGRATION

In order to enable haptic interaction, all objects modeled in the graphic workspace also need to be modeled on the haptic workspace. In order for users to *feel* what they actually *see*, the position of these 3D models needs to match and correspond on the scene. Graphic frames usually need to be rendered at 30fps to have a visually plausible effect. However, the suggested haptic update rate is of *1 KHz* [3]. On this matter, the number of deformable surface particles was kept low to maintain the stability of the system. In our system, the user interacts with the environment using an Omni Phantom [3] haptic device, as shown on Figure 1. Since there are differences between the graphic workspace and haptic workspace, the integration between graphic and haptic workspaces is required.

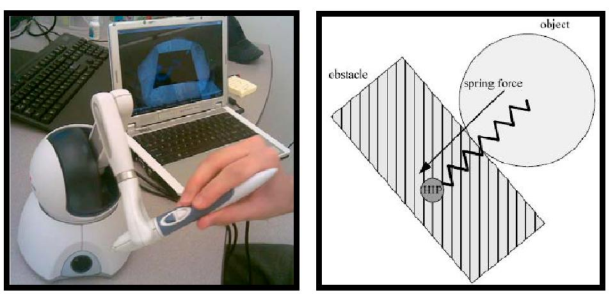

Fig. 1. Left image shows the Sensable Phantom Omni Haptic Device. Right thumbnail shows an illustration of the concept behind haptic force rendering.

We are interested in rendering immediate force feedback while maintaining acceptable visual simulation effects. As haptic interaction requires higher frequency updates, we must limit our system to a particular grid size in order to retain stability. However, the graphic workspace and the haptic workspace have different boundary limitations and coordinate systems. Therefore, the point of intersection between the haptic probe and the fluid surface is different in both workspaces. The fluid surface grid size is defined as an NXN square, and its coordinates range from $[0...N]$ in both X and Y axis of the deformable surface. If we want to know which part of the fluid surface was touched, we need to convert the haptic coordinates into those of the fluid grid. Therefore, we first convert the haptic coordinates into positive values, and then scale them with the graphic workspace boundaries. Based on these conversions, the graphic workspace and haptic workspace are integrated precisely and it shows performance in real-time. The 3D cursor represents the position of the probe as well as indicates the user's point of interaction.

We extended this integration with a 3D deformable surface. Our 3D deformable surface is designed based on a discreet mass-spring particle system [15]. The surface deforms with the touch of a haptic probe and gives back the resulting forces to the users. Even though higher resolutions of the surface grid provide smoother looks, our experiment shows that a size of *15X15* particles for the deformable surface is the maximum setup permitted to support a reasonable stable and fast real-time simulation. Once we increase the size, the deformable surface would perform too slowly for real-time purposes. This deformable layer is rendered at different heights to give the notion of 3D depth. The color interpolation is based on the density of the fluid cells. The higher the density, the brighter the color is.

FORCE RENDERING

In contrast with conventional haptic systems, our reaction force and torque feedback originate from two sources; (i) deformable surface – that accounts for elastic forces, and (ii) fluid simulation – provides values for *viscosity, density, velocity,* and *inertia*. At a basic level, a haptic device generates force-feedback based on the position of the probe's end-effector and the Haptic Interface Point (HIP). These two positions are initially the same, but as the player manipulates the haptic device, the HIP might traverse a collision surface. A force is then rendered at the haptic device which is directly proportional to the vector (times the stiffness scalar) between the device's end-effector and the position of the HIP. In Figure 1, the HIP's position has penetrated a static obstacle (e.g. the baton has touched a wall of the bowl). Since the end-effector cannot move to the HIP's position, a spring force is displayed at the haptic device and the users can feel a collision response.

Elastic spring forces are controlled by stiffness properties that are particular to rigid surfaces, like the walls of a bowl. However, when the probe enters an area of fluid, the force felt is that of a viscous force rather than a spring force. The fluid does not actually repel the probe, but just slows down its stirring movement, according to the density grid computed at the time. The higher the density contained at a grid cell, the harder it is to stir through it. The moving fluid also affects the position of the probe. If the fluid's velocity field is running to the left and the user tries to stir to the right, for instance, a higher force feedback will be felt until the velocity field has adapted itself to the new input forces. Following the law of inertia, the probe will remain in movement unless acted upon by an outside force.

The force feedback calculation is based on the equations of an incompressible Navier-Stokes fluid simulation, which enables the generation of forces and torques for use with haptic devices. The bowl can also be touched through the haptic interface, giving the player a sense of boundary limitations for the interaction. The deformable surface uses the classical fourth-order Runge–Kutta (RK4) method to solve the ordinary differential equations (ODE) formed by the applied forces and the constrained spring-network of particles.

The fluid surface is deformed as the haptic probe pushes through it. This deformation is gradually transmitted and damped to the lower layers based on their depth and fluid density. After a certain pop-through force threshold, the probe is able to penetrate the surface and interact with the inner 3D fluid. As a consequence, the sense of viscosity can be rendered in any direction of interaction as the probe moves on the three-dimensional scene.

In order to simplify and increase the stability of the haptic-graphic simulation, the fluid is contained in a constrained grid environment. The fluid may not tear apart nor spill over the container. Some drawbacks of this proposed method include limitations on the force values that can be rendered, tradeoffs on real-life physics, and restrictions on the grid size of the simulation.

V. RESULTS

Screenshots are shown in Figure 2. They present the fluid being stirred by the haptic device. We can see that the intensity of the color represents the density value of a cell. In addition, the fluid follows the velocity field that is being generated by the probe movements. In this manner, we can also perceive the wavy effect of the deformable surface. The same figure shows the velocity field that guides the movement of forces across the surface. After removing the probe out of the bowl, the fluid keeps moving itself and gradually reduces its waviness. In the same manner, if the user lets loose of the haptic device while still inside the fluid, the probe will continue to follow the flow's path.

Multiple layers of the 2D fluid surface were rendered at different heights. This enhances the 3D perception of depth into the scene. Performance is maintained at speed since the simulation is not volumetric but rather based on discreet extensions of two-dimensional layers.

In order to better appreciate the quality of haptic rendering, the user is able to mix and toggle between different force rendering modes. Each of these modes may be enabled or disabled according to the user's preferences. These force modes focus on particular aspects of the force feedback. The deformable-surface mode enables the user to feel the ripples on the fluid surface. The viscosity mode challenges the user to move around dense fluid. The flow mode guides the haptic probe, hence the user's hand, through the velocity field so that the user perceives the formed currents. The flow-resistant mode enables the user to modify the velocity field by applying forces that resist the current flow. As a result, the system serves as an experimental framework to analyze haptic experiments for fluid simulations.

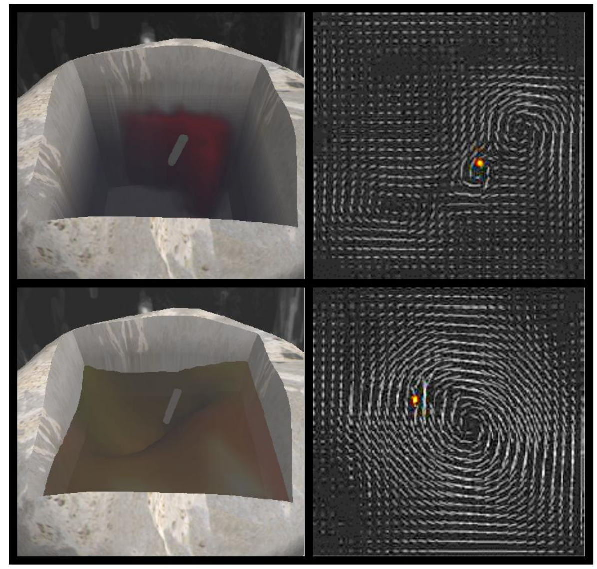

Fig. 2. These screenshots show the fluid being stirred by the haptic probe. The different levels of color represent the different levels of fluid density. Velocity grids are shown on the right.

Forces can also be visually appreciated by looking at the color of the baton during the simulation. These colors are dynamically modified according to the rendered forces. Users are able to associate the physical force feeling with the visual cue: a green shaded baton for light forces, a yellow color for moderate forces, and red for strong forces.

VI. ENHANCED USER-INTERACTION APPLICATION – GESTURE RECOGNITION

This section describes how this haptic fluid interactive system can be enhanced by integrating it with gesture recognition. A possible application can be a game situation such as the player impersonates the role of a witch. Following a specific recipe, a magic potion needs to be created. It would require the right ingredients, mixed at the right moment, with the proper stirring movements and force. Once the player succeeds, the system is able to trigger customized modifications to the fluid properties. The fluid might change color, viscosity and elasticity parameters among other characteristics. The decision might be made through a haptic motion recognition module as one of possible ways that will allow game developers to take full advantage of the high degree-of-freedom input capabilities of modern haptic devices.

Haptic devices provide more valuable parameters (force, torque, velocity, etc.) than conventional graphics users interfaces. It does not only allow us to recognize 3D coordinates, but also to use force-feedback data as extractable features. These parameters are used to raise the recognition rate of user motions. For instance, a harsh circular movement will be recognized differently than a gentle circular movement. Even though these are both circular movements, different forces were applied.

Haptic biometric behavioral applications [16] show the importance of force and torque for the purpose of recognition. We present how to recognize a few simple figures, also known as gestures, which would trigger the right potion spell, for instance, three consecutive circular motions or the shape of a star. This would reduce the complexity of the task, and therefore it would be more feasible for the recognition to be performed in real-time, parallel to the haptics and graphics fluid simulations.

This module follows Dopertchouk's concepts of motion recognition [17] and is organized in three major steps: Creation and storage of the master gesture templates, normalization of the strokes, and recognition of the shapes. The gesture templates are recorded from predefined sample haptic inputs. We read the proxy position of the haptic device and store the 4D coordinates as a sequence of points in the workspace, mainly *x, y, z* and *force* data. When players stir the potion mix, some of the gesture shapes may be different in size, speed, or position. Even thought the shape results might look similar to the naked eye, these shapes would look like completely unrelated sets of coordinates to the computer. Therefore, we need to normalize the captured strokes. First, we need to scale the gesture to a predetermined size (e.g. *1 unit*). We do this by finding the length and dimensions of the gesture, and dividing its coordinates by the scaling factor. Second, we need to put the individual points in the gesture at a uniform distance. We do this through a dynamic time warping algorithm [16]. Since we are interested in the geometric shape, it would be irrelevant to know how fast or slow the gesture was drawn. Finally, we need to center the gesture at the origin of the coordinate system through a translation matrix. We compare two different approaches for the haptic recognition of the gestures: Dot Product and Neural Network.

Simple shape recognition was performed through the implementation of a neural network-based recognition engine, using an approach similar to others [22][23], whom also provide good introductions to neural networks. A similar feature extraction procedure was used. However, haptic proxy positions were converted to directional vectors (e.g. Right: *1,0,0; 1,0,0; ...*). A back-propagation algorithm was used to train the neural network with a few basic shapes, run as many epochs and find the minimum sum-of-squares error (SSE) [21] constraint. However, this method proved to be cumbersome to perform in respect to the additional marginal benefit that we would get in the recognition phase. It would be more time-consuming to integrate a new predefined gesture into the system, as the network would need to be retrained. Therefore a simple Dot Product method is chosen for our recognition system.

Since both our gesture templates and captured strokes have the same number of points after normalization, we model our gestures as normalized vectors. These are *4XN* dimensional vectors, where *N* is the number of points in the gesture. Using this technique, if you compare two normalized vectors that are exactly the same, the result will be *one*. The result will be a value slightly less than *one* for vectors that point in more-or-less the same direction, and the result will be a low number for vectors that point in different directions. This worked well for simple shape matching and it didn't slow down any of the haptic fluid nor the deformable surface computations.

From a set of three basic motions (e.g., circle, *V* shape, and *S* shape), this module was able to reach recognition rates above *95%* for both recognition approaches. In our game scenario, the dot product approach seemed effective enough to recognize potion shapes. Neural networks also provided acceptable gesture recognition rates, but the time allocated for network retraining is cumbersome and tedious for gaming purposes.

VII. CONCLUSION

We have shown a novel human-computer system based on haptic-fluid interaction. It is the fluid simulation with the deformable surface together with the force feedback of the haptic device that requires very high-speed interaction rate. The system is stable and efficient. In addition, the realistic looking fluid rendering and haptic feedback have been successfully achieved. The system imitates the resulting physical forces generated by the stirring movements and fluid density changes. Our main contribution is to extend the Human-Computer interaction into fluid animation with force feedback. These fluid interaction techniques with haptic feedback have wide possible applications including game development and haptic communities. Haptic gesture recognition, as an application for haptic games, was experimented. A demonstration video of the system is available on the web [27].

OpenGL was used to implement the graphic framework of the system. We made use of lighting, blending, and shading effects to appreciate the animated fluid ripples. A bowl model was created on Autodesk 3D Studio Max [18] and imported into the scene. OpenHaptics API was used to model the haptic interactivity of the tool. The system performed on an Intel Pentium powered processor with *2GB* in RAM. A Sensable Phantom Omni [19] device was used as our haptic device.

It would be interesting to keep exploring the haptic gesture recognition phase of the project to produce more various effects on the fluid with game scenario. The orientation and workspace of the Phantom series of haptic devices allow the users to make natural, human gestures using a stylus. Even the idea of waving a haptic stylus through the air in order to cast spells is appealing in that it makes the player feel as if they really are wizards. This is a feature-in-progress, but current results look promising.

REFERENCES

[1] Stam, J.: Real-Time Fluid Dynamics for Games. In: Proceedings of the Game Developer Conference. March (2003)

[2] Srinivasan M., Basdogan C.: Haptics in Virtual Environments: Taxonomy, Research Status, and Challenges. In: Computer and Graphics, 21(4), pp. 393-404, (1997)

[3] Basdogan C.: Haptic rendering tutorial, http://network.ku.edu.tr/~cbasdogan/Tutorials/haptic_tutorial.html

[4] Ho C.H., Basdogan C., Srinivasan M.: Efficient point-based rendering techniques for haptic display of virtual objects. Teleoperators and Virtual Environments, pp.477-491, (1999)

[5] W. Mark, S. Randolph, M. Finch, J. V. Verth, R. M. Taylor II.: Adding Force Feedback to Graphics Systems: Issues and Solutions. SIGGRAPH'96, pp. 447-452, August (1996)

[6] S. Andrews, J. Mora, J. Lang, W.S. Lee: HaptiCast: A Physically-Based 3D Game with Haptic Feedback. In Proceedings of FuturePlay (2006)

[7] Morris D., Neel J., Salisbury K.: Haptic Battle Pong: High-Degree-of-Freedom Haptics in a Multiplayer Gaming Environment. In Experimental Gameplay Workshop, GDC (2004)

[8] Nilsson D., Aamisepp H.: Haptic hardware support in a 3D game engine. Master thesis, Department of Computer Science, Lund University, May (2003)

[9] Crystal Space 3D, http://www.crystalspace3d.org/

[10] Conti F., Barbagli F., Morris D., Sewell C.: CHAI: An Open-Source Library for the Rapid Development of Haptic Scenes. Paper presented at IEEE World Haptics, Italy, (2005)

[11] W. Baxter, M. C. Lin: Haptic Interaction with Fluid Media. Proceedings of Graphics Interface. Vol. 62, pp.81-88, Canada (2004)

[12] K. Lundin, M. Sillen, M. Cooper, A. Ynnerman: Haptic visualization of computational fluid dynamics data using reactive forces. In: Proceedings of the International Society for Optical Engineering, (2005)

[13] Stam J.: Stable Fluids. In: SIGGRAPH 99 Conference Proceedings, Annual Conference Series, pp.121-128. (1999)

[14] Stam J.: Interacting with smoke and fire in real time. In: Communications of the ACM, Volume 43, Issue 7, pp.76-83. (2000)

[15] Nealen A., Muller M., Keiser R., Boxerman E., Carlson M.: Physically based deformable models in computer graphics. In Eurographics: State of the Art Report. (2005)

[16] M. Orozco, A. El Saddik: Haptic: The New Biometrics-embedded Media to Recognizing and Quantifying Human Patterns. In proceedings of 13th Annual ACM International Conference on Multimedia (ACM-MM 2005), Singapore, November 06-12, (2005)

[17] O. Dopertchouk: Recognition of Handwritten Gestures, http://gamedev.net/reference/articles

[18] Autodesk, http://autodesk.com

[19] Sensable Technologies, http://sensable.com

[20] Logitech Rumblepad Gamepad, http://logitech.com

[21] C. M. Bishop: Neural Networks for Pattern Recognition, Oxford Press, USA, (1995)

[22] Boukreev K.: Mouse gestures recognition, http://codeproject.com/cpp/gestureapp.asp

[23] Murakami K., Taguchi, H.: Gesture Recognition using Recurrent Neural Networks. In Proceedings: Conference on Human Factors in Computing Systems, pp.237 - 242, (1991)

[24] Y. Dobashi, M. Sato, S. Hasegawa, T. Yamamoto, M. Kato, T. Nishita: A Fluid Resistance Map Method for Real-time Haptic Interaction with Fluids. Proceedings of ACM VRST'06. pp. 91-99 (2006)

[25] F. Dachille, H. Qin, A. Kaufman: Novel haptics-based interface and sculpting system for physics-based geometric design. Computer-Aided Design, Vol. 33, pp.403-420 (2001)

[26] Nintendo Wii, http://wii.com

[27] Haptic Fluid Demo Video, http://www.youtube.com/watch?v=DLM5HtnV5mY