Enabling design and interactive selection of haptic modes

Karljohan Lundin (Palmerius), Matthew Cooper, Anders Persson, Daniel Evestedt and Anders Ynnerman

Enabling Design and Interactive Selection of Haptic Modes

Karljohan Lundin[1], **Matthew Cooper**[1], **Anders Persson**[2], **Daniel Evestedt**[3], **Anders Ynnerman**[1]

[1] Norrköping Visualization and Interaction Studio, Linköping University, Sweden, e-mail: `{karlu,matco,andyn}@itn.liu.se`
[2] Centre for Medical Imaging and Visualization, Linköping University, Sweden, e-mail: `anders.persson@cmiv.liu.se`
[3] SenseGraphics AB, e-mail: `daniel.evestedt@sensegraphics.com`

Received: 13 July 2005 / Accepted: 2 May 2006

Abstract The ever increasing size and complexity of volumetric data in a wide range of disciplines makes it useful to augment volume visualization tools with alternative modalities. Studies have shown that introducing haptics can significantly increase both exploration speed and precision. It is also capable of conveying material properties of data and thus has great potential to improve user performance in volume data exploration.

In this paper we describe how recent advances in volume haptics can be used to build haptic modes — building blocks for haptic schemes. These modes have been used as base components of a toolkit allowing for more efficient development of haptic prototypes and applications. This toolkit allows interactive construction, configuration and fine-tuning of both visual and haptic representations of the data. The technology is also used in a pilot study to determine the most important issues and aspects in haptic volume data interaction and exploration, and how the use of haptic modes can facilitate the implementation of effective haptic schemes.

Key words volume haptics – haptic modes – toolkit – user study

1 Introduction

Volume visualization is rapidly becoming an indispensable tool in the analysis of the vast amount of information contained in volumetric data. The development of efficient tools that will support the analysis and filtering of this data continues to pose research challenges. Evaluations of simple, well defined tasks[1,2,3,4,5,6] have shown that haptics has the potential to significantly increase both speed and accuracy of human-computer interaction. Our sense of touch and kinaesthetics is also capable of providing large amounts of information about the location, structure, stiffness and other material properties of objects, that can be hard to represent visually.

Implementing effective haptic interaction for volume exploration is a non-trivial task, however. There are no established guidelines, generally little knowledge on how the haptic feedback should be integrated into the exploration process and limited support for volume haptics from the available software packages. We believe that an important reason for this is that the available methods have so far been unsuitable for general handling of volume haptics. Recent methods, introducing haptic primitives[7] provide a foundation suitable both for development of general systems and for formalization of haptic interaction with volumetric data.

In the first part of this article we describe how the haptic primitives are used to construct *haptic modes*, high-level haptic interaction definitions. To further facilitate the current developments and research, we present a toolkit for haptic volume visualization. In the second part of the article, we use the technology in a formative pilot study executed to motivate the formalization of haptic interaction and identify important aspects and requirements on volume haptics in volume data interaction and exploration.

The main contributions of this paper are:

- a formal definition of haptic modes as building blocks for implementing haptic schemes from haptic primitives
- the presentation of a toolkit for haptic volume exploration realizing the concept of a multi-modal data flow pipeline using haptic modes as building blocks
- the presentation of a real-time environment for interactive selection and configuration of haptic modes and visualization of volumetric data built on top of the toolkit
- a pilot study and analysis of how potential users interact with the haptic exploration of volume data, motivating design choices of the toolkit and providing recommendations for the design of haptic interaction

The next section gives an overview of the main approaches used in haptic interaction with volumetric data, their limitations and the haptic primitives approach that overcome these limitations. Section 3 then describes how this method is used to build the haptic modes and in section 4 we present the toolkit. In section 5 we present the interactive environment and in section 6 we describe the user study followed by conclusion in section 7.

2 Previous Work on Volume Haptics

Generating haptic feedback from surface data is a well-known field. Volumetric data, on the other hand, has no surface information from which haptic feedback can be generated. One way to overcome this problem is to extract surface data using thresholding in a pre-processing step, or to extract an intermediate local surface in real-time. Surface models for haptics, however, only represent a potential subset of the features in volumetric data. They also suffer from the occlusion of important areas by the use of distinct, impenetrable surfaces. To produce haptic feedback not limited to a predefined subset, a direct volume haptics approach is needed; a method that renders a continuous haptic information field throughout the entire volume, analogous to direct visual volume rendering.

There are two main approaches for direct volume haptics: force function-based and proxy-based. In most cases neither of these methods attempt to mimic the feedback found in the real world but rather aim at non-realistic perceptualization of the data for better intuitivity and effectiveness. The following two sections give a brief overview of the two approaches and describe the main problems that have limited the use of volume haptics. Section 2.3 then describes the haptic primitives[7] that enable the general approach used in our toolkit.

2.1 Force Function-based Volume Haptics

A simple and popular way to introduce haptic feedback from volumetric data is to define the force feedback as a vector valued function of the data around the haptic probe[4, 8, 9, 10, 11, 12], the active point of the haptic device. Common variables in a force function are the probe velocity, to simulate viscosity, and the gradient vector of the scalar field. The force function is then given as

$$\mathbf{F} = -C_1\,\mathbf{v}_\mathrm{p} - C_2\,\boldsymbol{\nabla} V(\mathbf{x}_\mathrm{p}) \tag{1}$$

where $\mathbf{v}_\mathrm{p}$ is the probe velocity, $V(\mathbf{x}_\mathrm{p})$ is the value of the scalar field at the probe position, $\mathbf{x}_\mathrm{p}$, and C_1 and C_2 are positive scalar constants or functions of some property of interest in the data. The viscosity term can be used to convey the scalar value by letting C_1 be a function of the scalar value at the probe position, and the gradient term can convey the orientation of the data surrounding the haptic probe by pushing the instrument towards high or low scalar values, depending on the sign of C_2.

In interactions with *vector data* the force functions can be as simple as using the interpolated vector value at the probe position[4]. A more advanced force function for vector data interaction could pull the haptic probe towards the core of vortex streams in the data[11, 12],

$$\mathbf{F} = -C_1\,(\boldsymbol{\nabla}\times\mathbf{V}(\mathbf{x}_\mathrm{p}))\times\mathbf{V}(\mathbf{x}_\mathrm{p}). \tag{2}$$

Force functions are easy to implement, and therefore quite popular. Representing data with a simple “pushing force” may, however, in some cases be too simplistic an approach since only certain data, such as pressure and flow, is convincingly represented by a simple force. In general, force functions provide a limited set of possible feedback. Furthermore, force functions tend to become unstable around areas of rapidly changing force direction.

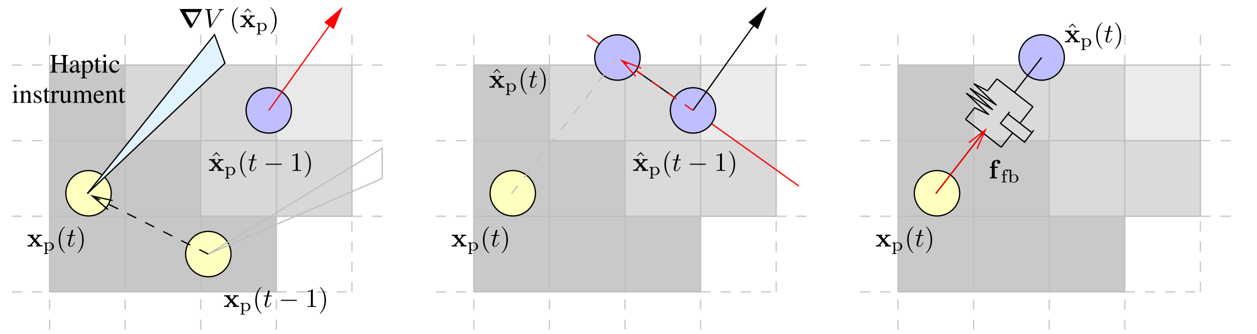
2.2 Proxy-based Volume Haptics

The notion of proxy-based volume haptics was introduced in [13] as a means to generate surface-like feedback from scalar density data without introducing haptic occlusion or limiting the interaction to a particular iso-value. The method has been used not only with scalar data, but also to render alternative shapes from vector[14, 15] and tensor[14] fields. Proxy-based approaches use a decoupling scheme where the probe has two representations: the probe, $\mathbf{x}_\mathrm{p}$, the haptic device that can only be directly affected through force feedback, and an internal proxy, $\hat{\mathbf{x}}_\mathrm{p}$, which is fully controlled by the algorithm. The feedback is then obtained, for each time-frame of the haptic loop, through the following three steps (also shown in Fig. 1): 1) collecting data at the local position being probed, 2) using the data to specify proxy movements, and 3) calculating force feedback from the proxy displacement relative to the probe.

1. Data properties First the data properties to be rendered by haptic feedback are extracted at the proxy position, for example the gradient vector (see Fig. 1(i)). To produce more fine tuned feedback from the volumetric data, a haptic transfer function, $\tau : \mathcal{R} \rightarrow \mathcal{R}$, can be used to map the data property to the strength of the physical force. This allows for representation of material properties[13, 9, 16] or gives the ability to emphasize specific features in the data in a manner analogous to the way visual transfer functions are used in volume rendering. Some examples of properties that are estimated and affect the haptic feedback are viscosity, friction, stiffness and flow strength.

2. Moving the proxy Simple one dimensional *constraints* are defined as functions of the material properties of the local data. Each constraint controls the movement of the proxy in one direction to constrain the movement of the haptic probe in that direction. By combining three independent orthogonal constraints in a local frame of reference, a feeling of surfaces, friction, viscosity or transverse damping can be generated. Thus, since the orientation of the frame and the strength of the constraints are controlled by the local data, the feedback reflects the shape and properties of the data. The proxy movement is calculated separately in each direction and combined linearly to give the new proxy position, see Fig. 1(ii).

3. Calculating feedback After the new proxy position has been determined the force feedback is calculated using a virtual spring-damper system, coupling the probe to the proxy,


$\nabla V(\hat{\mathbf{x}}_{\mathrm{p}})$
Haptic instrument
$\hat{\mathbf{x}}_{\mathrm{p}}(t-1)$
$\mathbf{x}_{\mathrm{p}}(t)$
$\mathbf{x}_{\mathrm{p}}(t-1)$
$\hat{\mathbf{x}}_{\mathrm{p}}(t)$
$\hat{\mathbf{x}}_{\mathrm{p}}(t-1)$
$\mathbf{x}_{\mathrm{p}}(t)$
$\hat{\mathbf{x}}_{\mathrm{p}}(t)$
$\mathbf{f}_{\mathrm{fb}}$
$\mathbf{x}_{\mathrm{p}}(t)$

(i) Step 1: Evaluate local data properties, in this example the gradient vector, $\nabla V(\hat{\mathbf{x}}_{\mathrm{p}})$, at the proxy position, $\hat{\mathbf{x}}_{\mathrm{p}}$.

(ii) Step 2: Move the proxy point according to haptic constraints, in this example to simulate a plane.

(iii) Step 3: Calculate feedback force by simulating spring-damper coupling.

Figure 1 Three steps for generating proxy-based haptic feedback, in this example from a virtual surface.

see Fig. 1(iii). Thus the force feedback $\mathbf{f}_{\mathrm{fb}}$ is evaluated through

$$\mathbf{f}_{\mathrm{fb}} = k\left(\hat{\mathbf{x}}_{\mathrm{p}} - \mathbf{x}_{\mathrm{p}}\right) + D\left(\hat{\mathbf{v}}_{\mathrm{p}} - \mathbf{v}_{\mathrm{p}}\right) \tag{3}$$

where $\hat{\mathbf{x}}_{\mathrm{p}}$ and $\mathbf{x}_{\mathrm{p}}$ are proxy and probe position, $\hat{\mathbf{v}}_{\mathrm{p}}$ and $\mathbf{v}_{\mathrm{p}}$ are proxy and probe velocity, and k and D are stiffness and damping parameters.

The above outlined constraint approach to proxy-based volume haptics use an *orthogonal frame of constraints* to produce feedback. The method is incapable of handling *non-orthogonal* constraints correctly, as is shown in [17]. The orthogonal requirement is far too restrictive and for a more general applicability and larger set of possible haptic schemes we have to address this problem.

2.3 Haptic Primitives

Haptic primitives for proxy-based volume haptics were introduced in [7]. We describe them briefly here again since they constitute a foundation for the rest of this paper. For more details please refer to the earlier publication.

Haptic primitives both form a comprehensive abstraction layer for the implementation of haptic interaction schemes and provide an effective means of calculating the feedback. Constraints are represented using primitives of one, two and three degrees of freedom: *plane*, *line* and *point*, respectively. Active forces and other force functions are included through a fourth primitive: *directed force*. The effects of the primitives are illustrated in Fig. 2. Superpositions of these four primitives are sufficient to represent any force feedback scheme[7]. In addition this method avoids the requirement of orthogonal constraints that is a persistent problem with the proxy-based approach[13, 14, 15].

Each haptic primitive is characterized by a simple force equation, with parameters strength, s, position, $\mathbf{x}$, and direction, $\mathbf{q}$:

- Directed force, a position-independent force:

$$\mathbf{F}_i\left(\hat{\mathbf{x}}_{\mathrm{p}}\right) = s_i \mathbf{q}_i \tag{4}$$

- Point, an attractor to a point in space:

$$\mathbf{F}_i\left(\hat{\mathbf{x}}_{\mathrm{p}}\right) = \begin{cases} \mathbf{0}, & \text{if } |\mathbf{x}_i - \hat{\mathbf{x}}_{\mathrm{p}}| = 0 \\ s_i \frac{\mathbf{x}_i - \hat{\mathbf{x}}_{\mathrm{p}}}{|\mathbf{x}_i - \hat{\mathbf{x}}_{\mathrm{p}}|}, & \text{if } |\mathbf{x}_i - \hat{\mathbf{x}}_{\mathrm{p}}| \neq 0 \end{cases} \tag{5}$$

- Line, an attractor towards the closest point on a line:

$$\mathbf{F}_i\left(\hat{\mathbf{x}}_{\mathrm{p}}\right) = \begin{cases} \mathbf{0}, & \text{if } |\mathbf{m}| = 0 \\ s_i \frac{\mathbf{m}}{|\mathbf{m}|}, & \text{if } |\mathbf{m}| \neq 0 \end{cases} \tag{6}$$

  where $\mathbf{m}$ is the vector to the closest point on the line defined by $\mathbf{x}$ and $\mathbf{q}$, $\mathbf{m} = \mathbf{q}_i\left[\mathbf{q}_i \cdot \left(\hat{\mathbf{x}}_{\mathrm{p}} - \mathbf{x}_i\right)\right] - \left(\hat{\mathbf{x}}_{\mathrm{p}} - \mathbf{x}_i\right)$.
- Plane, a directed force which exists only on one side of the plane:

$$\mathbf{F}_i\left(\hat{\mathbf{x}}_{\mathrm{p}}\right) = \begin{cases} 0, & \text{if } \left(\hat{\mathbf{x}}_{\mathrm{p}} - \mathbf{x}_i\right) \cdot \mathbf{q}_i \geq 0 \\ s_i \mathbf{q}_i, & \text{if } \left(\hat{\mathbf{x}}_{\mathrm{p}} - \mathbf{x}_i\right) \cdot \mathbf{q}_i < 0 \end{cases} \tag{7}$$

The proxy position for each time frame is then found by balancing the force feedback from the coupling equation against the force from the primitives, by numerically minimizing the residual ε in

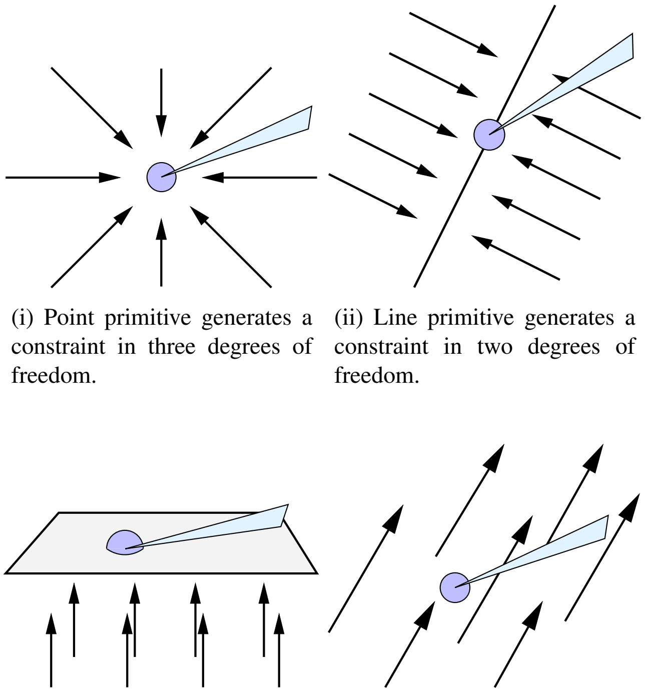
$$\varepsilon = -\mathbf{f}_{\mathrm{fb}}\left(\hat{\mathbf{x}}_{\mathrm{p}}\right) + \sum_i \mathbf{F}_i\left(\hat{\mathbf{x}}_{\mathrm{p}}\right) \tag{8}$$

with respect to $\hat{\mathbf{x}}_{\mathrm{p}}$. All other primitive parameters, such as the primitive positions, are held constant.

To simplify the expressions of ε in the following section we let $\odot^s_{\mathbf{x}}$, $\oslash^s_{\mathbf{q},\mathbf{x}}$, $\obar^s_{\mathbf{q},\mathbf{x}}$ and $\ominus^s_{\mathbf{q}}$ represent the force functions for the point, line, plane and directed force primitives, respectively.

3 Haptic Modes

Using the primitives, schemes for haptic interaction can be handled as entities, *haptic modes*, simplifying the design and implementation of visio-haptic interfaces. In this section we describe the relationship between the haptic primitives and the implementation of haptic modes.



(i) Point primitive generates a constraint in three degrees of freedom.

(ii) Line primitive generates a constraint in two degrees of freedom.

(iii) Plane primitive generates a constraint in one degree of freedom.

(iv) Directed force pushes the haptic instrument in a direction.

Figure 2 The four haptic primitives and their effects. Haptic schemes are implemented by selecting haptic primitives and controlling their parameters, such as the orientation, as functions of the local data.

The haptic modes are implemented by selecting haptic primitives and controlling their parameters, such as the orientation, as functions of the local data. In that respect the haptic mode acts as a link between data and its haptic representation. Since the approach allows haptic primitives to be freely combined, so can the haptic modes. Thus, a large set of fairly simple haptic modes can be used, both individually or combined into more advanced haptic schemes. Below a set of haptic modes is described. Some of these modes also provide typical examples of each haptic primitive; its effect and use.

The primitives, selected to generate the feedback for a haptic mode, are placed at the location of the proxy point from the previous time-frame. Thus, the primitives generate local haptic shapes at any probed position in the volume, which produces a smooth continuous representation of the data. By locating a primitive at other positions, alternative effects can be provided, for example a snap-drag effect[14].

Viscosity mode Viscosity can be simulated by adding an attraction towards the position where the proxy point was located in the previous time-frame, $\hat{\mathbf{x}}_\mathrm{p}$. This will introduce a braking force on the proxy. For this we use a point primitive placed at the old proxy position. To produce velocity-based viscosity, the strength of the primitive may be defined as a function of the proxy velocity. Using the velocity, however, makes the feedback dependent on the speed of exploration as well as the scalar value of the data. To avoid that we choose to control the strength using a transfer function, τ_visc, of the scalar value at the proxy position, as discussed in section 2.2. The residual to minimize is then expressed as

$$\boldsymbol{\varepsilon} = -\mathbf{f}_\mathrm{fb} + \bigodot_{\mathbf{x}=\hat{\mathbf{x}}_\mathrm{p}}^{s=\tau_\mathrm{visc}(V(\hat{\mathbf{x}}_\mathrm{p}))} \tag{9}$$

Gradient force mode The gradient term in function-based haptic interaction, discussed in section 2, is useful when interacting with pressure or fluid data. A force from the gradient of a scalar dataset can easily be produced using the directed force primitive. We let the strength of the force be specified through a transfer function, τ_grad, from the magnitude of the gradient vector and use the normalized gradient vector, $\mathbf{n} = \frac{\boldsymbol{\nabla} V(\hat{\mathbf{x}}_\mathrm{p})}{|\boldsymbol{\nabla} V(\hat{\mathbf{x}}_\mathrm{p})|}$, as the direction of the force,

$$\boldsymbol{\varepsilon} = -\mathbf{f}_\mathrm{fb} + \mathop{\bigcirc\kern-0.9em\rightarrow}_{\mathbf{q}=\mathbf{n}}^{s=\tau_\mathrm{grad}(|\boldsymbol{\nabla} V(\hat{\mathbf{x}}_\mathrm{p})|)} \tag{10}$$

If the gradient magnitude is zero we use an arbitrary vector to maintain consistency. By setting the transfer function to zero for zero magnitude gradients this arbitrary vector does not contribute to the force feedback. This way of avoiding division by zero is also applied in the following modes.

Force mode The simple mapping between vector field and force, mentioned in section 2, is used in our user study, section 6. The mapping is implemented using a force primitive, so the residual is expressed as

$$\boldsymbol{\varepsilon} = -\mathbf{f}_\mathrm{fb} + \mathop{\bigcirc\kern-0.9em\rightarrow}_{\mathbf{q}=\boldsymbol{\nu}}^{s=\tau_\mathrm{force}(|\mathbf{V}(\hat{\mathbf{x}}_\mathrm{p})|)} \tag{11}$$

where τ_force is a transfer function and $\boldsymbol{\nu} = \frac{\mathbf{V}(\hat{\mathbf{x}}_\mathrm{p})}{|\mathbf{V}(\hat{\mathbf{x}}_\mathrm{p})|}$.

Vector follow mode In previous work, we have encountered haptics that guides the user to follow the vectors of vector fields[14, 18, 19]. This is useful when exploring flow data, such as heart MRI and data from Computational Fluid Dynamics (CFD). It guides the haptic instrument in the orientation of the vector field by presenting a resistance in directions orthogonal to the vector field. This mode is easily implemented using a line primitive. To orient the primitive the normalized vector value at the proxy position is used and the strength is defined by the vector length through a transfer function, τ_vec,

$$\boldsymbol{\varepsilon} = -\mathbf{f}_\mathrm{fb} + \mathop{\bigcirc\kern-0.8em/}_{\mathbf{x}=\hat{\mathbf{x}}_\mathrm{p},\,\mathbf{q}=\boldsymbol{\nu}}^{s=\tau_\mathrm{vec}(|\mathbf{V}(\hat{\mathbf{x}}_\mathrm{p})|)} \tag{12}$$

Vortex tube mode In a recent application[15] for exploring data from CFD we implemented a version of the vortex core mode described in section 2. This mode is implemented using a plane primitive with the vortex core direction of Eqn. 2, $\boldsymbol{\varphi} = (\boldsymbol{\nabla} \times \mathbf{V}(\hat{\mathbf{x}}_\mathrm{p})) \times \mathbf{V}(\hat{\mathbf{x}}_\mathrm{p})$, defining the orientation of the primitive. Thus haptic descriptions of the vortex shape and extension are generated rather than just of the vortex core. The strength of the primitive, and thus of the rendered tube, is determined through a transfer function, τ_tube, from the magnitude of the vector. This corresponds to the vorticity of the

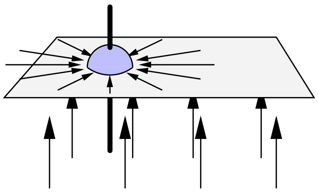


Figure 3 Surface and friction simulation by using plane and line primitives at the position of the proxy point of the previous time-frame.

vector field at the exploration point. The residual is then expressed as

$$\varepsilon = -\mathbf{f}_{\mathrm{fb}} + \mathop{\bigcirc\mkern-15mu\parallel}\nolimits_{\mathbf{x}=\hat{\mathbf{x}}_{\mathrm{p}},\,\mathbf{q}=\frac{\boldsymbol{\varphi}}{|\boldsymbol{\varphi}|}}^{s=\tau_{\mathrm{tube}}(|\boldsymbol{\varphi}|)} \tag{13}$$

Surface and friction The surface-and-friction feedback described in [13] can also be implemented using haptic primitives. A plane primitive oriented by the normalized gradient vector simulates surfaces. Since friction feedback is limited to two dimensions, that is the plane exerting the friction feedback, the friction is effectively simulated using a line primitive. The extension of the line primitive makes the friction effect consistent even if the surface is penetrated. This setup is shown in Fig. 3. The strength of the surface is defined by the scalar data using a transfer function, τ_{surf}. A friction force, however, is generally calculated from the normal force, which is not known until the residual has been minimized. We, therefore, use the strength of the normal force from the previous time-frame, $k\mathbf{n} \cdot (\hat{\mathbf{x}}_{\mathrm{p}} - \mathbf{x}_{\mathrm{p}})$ where $\mathbf{n}$ is the surface normal, as an estimate of the current normal force. This force must, however, not exceed the surface strength. The strength of the line primitive is then calculated by multiplying the normal force strength with a friction value obtained from a transfer function, τ_μ, so

$$\begin{aligned}\varepsilon = -\mathbf{f}_{\mathrm{fb}} &+ \mathop{\bigcirc\mkern-15mu\parallel}\nolimits_{\mathbf{x}=\hat{\mathbf{x}}_{\mathrm{p}},\,\mathbf{q}=\mathbf{n}}^{s=\tau_{\mathrm{surf}}(V(\hat{\mathbf{x}}_{\mathrm{p}}))} \\ &+ \mathop{\oslash}\nolimits_{\mathbf{x}=\hat{\mathbf{x}}_{\mathrm{p}},\,\mathbf{q}=\mathbf{n}}^{s=\tau_\mu(V(\hat{\mathbf{x}}_{\mathrm{p}}))\min(\tau_{\mathrm{surf}}(V(\hat{\mathbf{x}}_{\mathrm{p}})),\,k\mathbf{n}\cdot(\hat{\mathbf{x}}_{\mathrm{p}}-\mathbf{x}_{\mathrm{p}}))}\end{aligned} \tag{14}$$

Combined modes When two or more modes are used simultaneously their individual force function contributions are combined linearly, as expressed by Eqn. 8. As an example consider the combination of the surface and friction mode and the viscosity mode. The combined residual to minimize becomes

$$\begin{aligned}\varepsilon = -\mathbf{f}_{\mathrm{fb}} &+ \mathop{\bigcirc\mkern-15mu\parallel}\nolimits_{\mathbf{x}=\hat{\mathbf{x}}_{\mathrm{p}},\,\mathbf{q}=\mathbf{n}}^{s=\tau_{\mathrm{surf}}(V(\hat{\mathbf{x}}_{\mathrm{p}}))} \\ &+ \mathop{\oslash}\nolimits_{\mathbf{x}=\hat{\mathbf{x}}_{\mathrm{p}},\,\mathbf{q}=\mathbf{n}}^{s=\tau_\mu(V(\hat{\mathbf{x}}_{\mathrm{p}}))\min(\tau_{\mathrm{surf}}(V(\hat{\mathbf{x}}_{\mathrm{p}})),\,k\mathbf{n}\cdot(\hat{\mathbf{x}}_{\mathrm{p}}-\mathbf{x}_{\mathrm{p}}))} \\ &+ \mathop{\odot}\nolimits_{\mathbf{x}=\hat{\mathbf{x}}_{\mathrm{p}}}^{s=\tau_{\mathrm{visc}}(V(\hat{\mathbf{x}}_{\mathrm{p}}))}\end{aligned} \tag{15}$$

This less restricted handling of haptic modes has the potential to widen the possible applications of volume haptics and increase its availability.

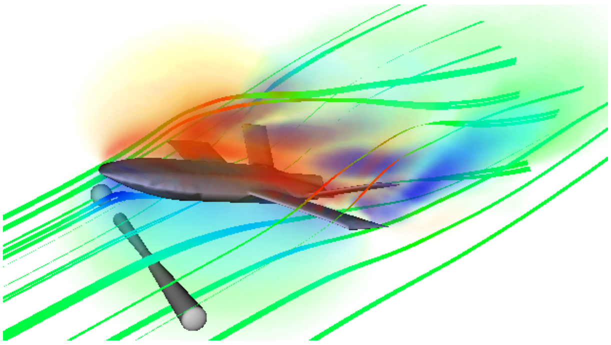


Figure 4 CFD dataset of an experimental UAV. In this visualization the air-flow field and the flow magnitude are rendered using stream-ribbons and volume rendering, respectively. The follow mode produces a haptic representation of the flow field while the vortex mode facilitates the exploration of the vorticity in the data.

4 Volume Haptics Toolkit

There are a number of systems currently available for general haptic rendering. To the authors knowledge, however, none exists that natively supports direct haptic interaction with volumetric data. Any example of direct volume haptics follows one of the two main approaches described above, deployed on a system primarily designed for surface haptics. Our volume haptics toolkit (VHTK) aims to meet this need for a framework for building applications for multi-modal volume data exploration. This section describes the toolkit and the measures taken to facilitate the design of haptic interaction.

The toolkit is implemented using H3D from SenseGraphics AB, extending it into the domain of volume visualization and haptics. H3D is a cross platform, open-source system based on the X3D standard. It has a common scenegraph for both haptics and graphics and provides, in addition to the standard graphical nodes, nodes for specifying haptic properties for geometries and to implement custom force models. The system is programmed at three different levels: X3D files are used to build scenes and set up simple dependencies between nodes with routes. For more complex behaviour and changes to the scenegraph the Python scripting language is used. Finally, C++ is used to create new nodes for haptics as well as for graphics and other tasks that require low-level programming.

VHTK enhances H3D by adding the nodes needed for producing haptic interaction with volumetric data including visual feedback, such as visualization nodes, data container nodes and data processing nodes. The nodes provided by the toolkit are implemented in C++ but can also be used and controlled from X3D and Python, allowing a programmer to either build the application in pure X3D and/or Python, or extend the toolkit further using C++. The toolkit has already been used in related projects, for example the fluid flow application shown in Fig. 4. It has also been released to public use under the GNU General Public License and can be downloaded from the H3D website (`http://www.h3d.org`).

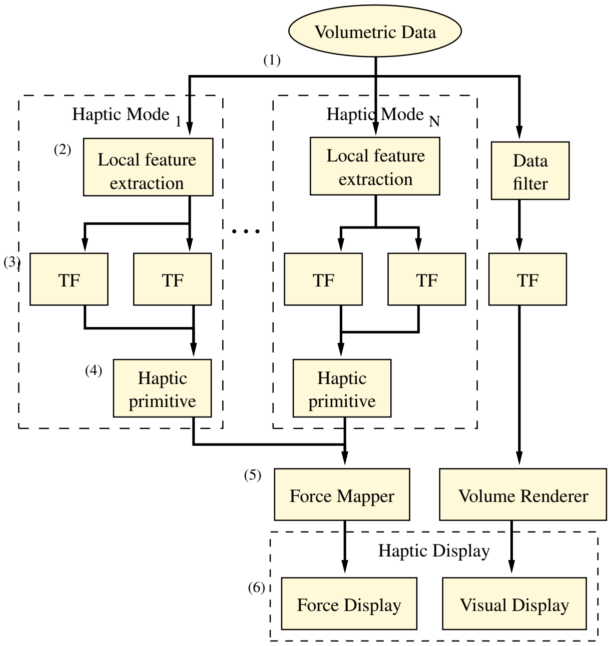


Figure 5 The conceptual data flow model of VHTK. The abbreviation “TF” denotes a transfer function.

4.1 Processing Pipeline

To provide both flexibility and real-time configurability we have designed the toolkit to support a highly configurable data flow model similar to that of rapid software implementation APIs, such as *The Visualization Toolkit* (VTK), see Fig. 5. The haptic modes, constructed from the primitives, are the building blocks that are treated as configurable filters in the data flow model. A typical multi-modal pipeline for generating haptic feedback from volumetric data is as follows:

1. volumetric data is read and trilinearly interpolated at the local position currently being explored by the haptic probe
2. simple data processing is used to extract local features from the volumetric data, for example the gradient vector or vector curl
3. scalar properties of the data, such as gradient magnitude, are fed through a transfer function to generate material properties, such as friction or surface strength
4. the material properties and data features are used to control the parameters of haptic primitives chosen to haptically represent the volumetric features
5. all haptic primitives are together mapped to a single force that represents the combined feedback, through the algorithm described in section 2.3
6. finally, typically at a rate of 1 kHz, the force derived from the primitives is exerted through the haptic instrument.

In steps 2–4 the choices of processing, transfer functions and types of the primitives affect the haptic behaviour, that is, they define the haptic mode.

The X3D standard defines an event handling and processing system to support dynamic behaviour and changes in the scenegraph. In this system a node can be made aware of the changes that are made in the data of a child node. This event propagation system is the basis for our conceptual data visualization pipeline. In the scenegraph a source is set as a child of the node using it. An example of this is shown in Fig. 6. If a parameter in a reader or a filter is updated this will trigger an event that propagates updates up the scenegraph and so through the data pipeline. This way the event propagation is also optimized to only perform time-consuming data processing when needed. It allows, in addition to fast and easy construction, real-time fine-tuning of new haptic interaction schemes.

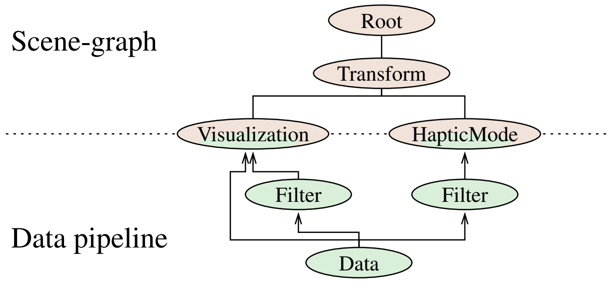


Figure 6 An example of how parts of the scenegraph are treated as a processing pipeline.

4.2 Haptic Nodes and Rendering

The toolkit encapsulates the steps forming the haptic behaviour into scenegraph nodes, thereby hiding the low-level processing and haptic primitives, see Fig. 5. The haptic nodes thus form a palette of modes that can be freely selected and combined to generate a wide array of different haptic schemes, allowing a developer to tailor the task specific haptic scheme of an application. Each node also provides an X3D interface to mode specific data and parameters. Analogous to visual models in visual scenegraphs, the transforms above the node affect the position and orientation of the haptic representation of the node’s data source. Letting haptic nodes and visualization nodes share parent transform and data source, as in the example shown in Fig. 6, thus provides co-located haptics and graphics.

Currently eight pre-implemented haptic modes are provided for representing features in both scalar and vector data. Some of the modes are described in section 3 and some are reimplementations of schemes presented in [18,12,14,4,15]. For some application areas and selected tasks the available predefined haptic modes, or combinations thereof, may not suffice to represent the most interesting features. If so, a fundamentally new haptic mode is needed. The low-level abstraction layer constituted by the haptic primitives is made available for the implementation of new haptic modes. New modes can easily be integrated into the toolkit framework by extending the abstract haptic node type and implementing the new node to provide haptic primitives describing the desired effect, as described in section 3.

4.3 Data Processing Scenegraph Nodes

Data sources and data filters are also implemented as scenegraph nodes. They provide a general data extraction interface for subsequent nodes to use and the filters differ from the other sources, such as readers, only in that their X3D interface allows the assignment of a source to read data from. The data handling structure allows for both analytical and sampled volume data and defines interfaces for extracting the basic features from scalar and vector data: scalar value, scalar gradient vector, vector value, vector curl vector and vector divergence. Among the filters provided by the toolkit are support for conversion between data types and extraction of the magnitude of vector features, such as vector curl, as scalar data for visual volume rendering or haptic feedback. By changing the filtering of the data used by a haptic mode its possible uses can be widely expanded. For example, in a related project a classification algorithm is used as filtering to enhance the haptic feedback.

To provide flexible and intuitive control of the scenegraph nodes, the toolkit makes extensive use of transfer functions. Filters for rescaling data use transfer functions to control the input/output conversion and both visual and haptic nodes use transfer functions to control material, colour and size properties. There are, therefore, several different types of transfer function nodes available, providing different control interfaces. Examples are specification of piecewise linear segment and using the window function common in radiology.

4.4 Visual Scenegraph Nodes

The main purpose of the toolkit is to provide an interface to advanced volume haptics, so only a few visualization nodes have, so far, been provided, such as volume rendering and stream-ribbons. However, more specialized packages for graphical visualization can also be used together with H3D and VHTK to provide more elaborate visual representations of the data.

5 Interactive Environment

To facilitate the process of building the haptic interaction scheme for a particular problem and demonstrate how dynamic applications can be implemented using VHTK, we have used the toolkit to build an interactive tool for deployment of haptic exploration. With this environment it is possible to load volumetric data, perform simple visualizations and interactively select and setup haptic modes from the palette provided by the toolkit.

5.1 Implementation

The graphical user interface (GUI) for the interactive environment is implemented in Tcl/Tk, through the Python module *Tkinter*. The interface runs from a Python script in the H3D scenegraph and the individual event handlers of Tk and H3D are connected through the Python interface.

In the design of the environment we group together each volume data reader with all the haptic modes that can be used on that type of data and the visual renderers for that type of data. All involved modes are automatically added to the scenegraph when a reader is selected, but, through our GUI, the user may turn on or off individual modes so that either one or several may be active at a time. Transfer functions, used to control the parameters of the modes, can also be adjusted through the interface. The transfer functions are drawn freehand, as can be seen in the example shown in Fig. 7. As the user updates the parameters the node configurations are updated in real-time, giving immediate feedback through the visualization and haptic interaction. The user can thus try out different haptic modes and different combinations and can also fine-tune the behaviour of the modes.

Vector data is visualized using interactive stream-ribbons provided by the `StreamRibbons` node. When this function is activated through the GUI, the user can interactively place ribbon seed points throughout the volume and move them with the resulting stream-ribbons being updated in real-time. Several parameters for the ribbons, such as colour transfer functions and ribbon length, can be controlled from the GUI.

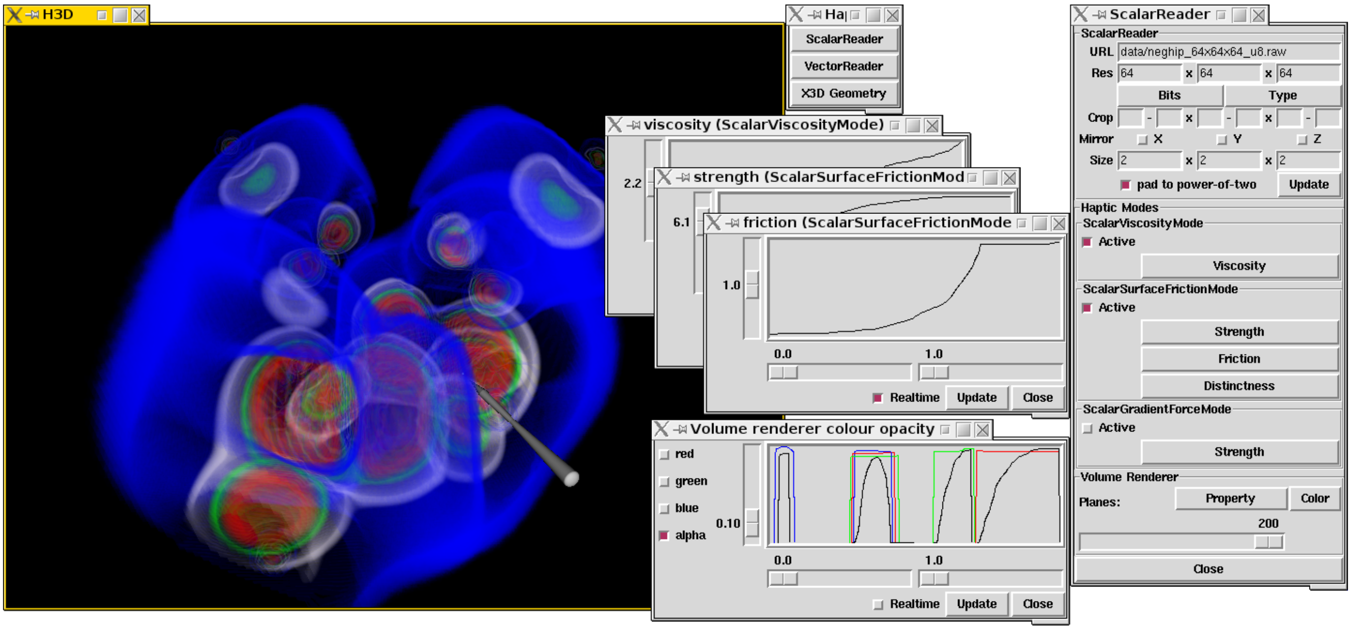
The volume renderer provided by the toolkit is used to visualize both scalar and vector data. Through a property extractor, also provided by the toolkit, the property to be rendered can be selected. For scalar data either the scalar data or the gradient magnitude is extracted and for vector data one of the magnitude of the vector, curl or divergence is extracted.

5.2 Interactive Example

As an example of how data can be explored using this environment, consider the simulation of the spatial probability distribution of the electrons in a high potential protein molecule. This data was obtained from the VolVis distribution of SUNY Stony Brook (see `http://www.volvis.org`). It is unsigned 8 bit integer data, 64^3 voxels in size. Using the volume visualization provided in the interactive environment for the scalar electron probability data we get the result shown in Fig. 7. The figure also shows the controls for selecting haptic modes and visualization, and fine-tuning material and colour properties. Currently activated are the viscosity mode and the surface-and-friction-mode, rendering a continuous set of surfaces from the probability density distribution. The haptic instrument can be freely moved into a high density region but when moving outwards a surface is perceived, conveying the shape of the local density distribution.

6 A Pilot Study on Volume Haptics

In this section we present a pilot study designed to review the effect of haptics in volume exploration. Many previous studies, for example [1,2,3,4,5,6], have successfully shown that haptic feedback can produce positive effects, by showing the



URL data/neghip_64x64x64_u8.raw
Res 64 x 64 x 64
Size 2 x 2 x 2
☒ pad to power-of-two

Figure 7 The graphical user interface: visualization window with stylus visible, list of readers, window for configuring a scalar volume reader, its haptic modes and the volume renderer that visualize the volume, and dialogs for real-time transfer functions. In this setup the viscosity mode is used to enhance the visual impression. The dataset is a simulation of the spatial probability distribution of the electrons in a high potential protein molecule, courtesy of VolVis distribution of SUNY Stony Brook (see `http://www.volvis.org`).

efficiency of adding haptic feedback to isolated sub-tasks. In contrast, we try to create an application scenario and so find how the potential user reacts to the new experiences introduced by haptic interaction and so deepen the understanding of *how*, and not *if*, volume haptics can assist in everyday tasks.

The primary aim of the study is to show how the formalization of haptic interaction facilitates the haptic design and thus justify the deployment of the new toolkit and validate the design choices. We do this by identifying important aspects and issues that need to be addressed by an application for visio-haptic volume visualization and interaction, in the context of haptic modes as a description of the haptic interaction. A secondary aim is to be able to provide suggestions for the implementation of future applications with visio-haptic interfaces for volume data exploration.

The study is a formative evaluation and follows the cooperative approach, an empirical technique common in HCI studies. The main basis for the study is conversations between subjects and an experimenter during a controlled case scenario. This is complemented by interviews and a short questionnaire. The study is designed to register subjective reactions and is, therefore, setup without specific questions. It is not expected to provide specific answers or be amenable to statistical significance, but rather give a deeper understanding of haptic interaction with volumetric data and indications and suggestions for future implementations and further studies.

6.1 Application and Case Setup

The case chosen for the pilot study is taken from medical visualization. With modern Magnetic Resonance Imaging (MRI) modalities, not only morphologic data but also fully 3D flow data can be acquired[20]. This provides a basis for research on, for example, the blood flow patterns in the human heart[21, 22]. The task in the current case is to explore the flow data and identify paths.

The MR scanner used to acquire the data produces two datasets, one scalar and one vector. The haptic interaction is derived from a 3×32 bit floating point vector dataset of $120 \times 90 \times 30$ voxels. The scalar dataset is of the same resolution and used for the visual volume rendering. The method used to acquire fully 3D flow information, however, produces poor tissue contrast, so the visual quality of the dataset is limited even though pre-processing has been applied to enhance the contrast, see Fig. 8. This makes understanding the flow information crucial for the clinicians, since they can not rely on the morphological information.

Since this study is neither a comparison nor competition between visual and haptic rendering, we feel no need for more advanced visualization than standard techniques. We use classical volume rendering techniques to visualize the heart morphology and the blood flow data is visualized using interactive stream-ribbons. Each ribbon can be placed at any position and moved, in real-time, through the volume using the haptic interaction device.

In this application we make use of a combination of three different haptic modes: the follow mode, the force mode and the gradient mode. We use the follow mode to convey the blood flow orientation. It guides the user to follow the field

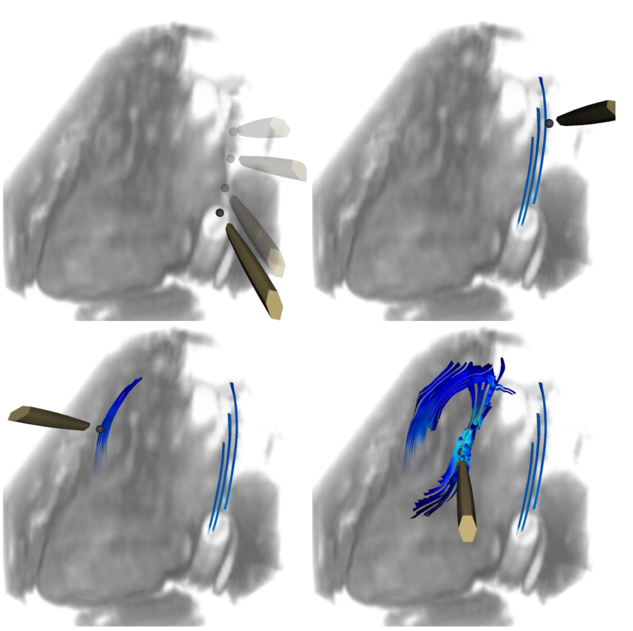


Figure 8 Identifying blood flow in haptic exploration. The haptic feedback makes it easier to find and follow the blood flow in the dataset. The bad tissue contrast caused by the pulse sequence used when acquiring the vector data makes the sense of touch even more important in the exploration process.

orientation and also gives a feeling of both the orientation of the local flow and of the strength of the flow, through the strength of the feedback. The force mode is applied to the flow data. Although this feedback is not identical to real flow drag, this pushes the haptic instrument in the direction of the flow and thus conveys the flow direction to the user. In a preprocessing step we also generate a scalar field from the vector magnitude and on this the gradient mode is applied. Since the gradient points towards stronger flow, this mode will produce a gentle pull towards regions with fast blood flow. Together these haptic modes are used to convey both the orientation and direction of the blood flow combined with a sense of where the blood flow is stronger. This is anticipated to help the users to understand the data and guide them to the main flow patterns, see Fig. 8. The underlying techniques, i.e. haptic modes and primitives, also allows individual modes to be activated and deactivated during run-time.

6.2 User Study

Seven experienced radiologists participated as test subjects, six with prior experience of 3D medical visualization and one without. Each radiologist was invited to a private supervised session following four steps:

1. introduction to haptics,
2. exploration of synthetic training data,
3. exploration of blood flow data, and
4. interview, questions and discussion.

The training data was used to familiarize the subjects with the interface and make them understand the nature of the haptic feedback. All three haptic modes described above were demonstrated. The exploration of the authentic blood flow data was the main part of the session. Each subject was given the task of finding and marking blood flow paths in the heart dataset using stream ribbons — a task easy to describe but difficult in practice. To focus on the haptic interaction, the user interface was simplified by using preset visual and haptic settings.

The subjects' experiences were discussed during four phases of the exploration:

1. using stream-ribbons without haptics,
2. using haptics without stream-ribbons,
3. using haptics together with stream-ribbons, and
4. using stream-ribbons without haptics again.

During exploration with haptic feedback, the follow mode was set as the default haptic scheme and the other two modes were turned on or off at the subjects' request.

The subjects were asked to continuously describe and discuss their experiences during the session. When needed the supervisor asked general open questions to trigger the subject to describe the experience. At the end of the session a questionnaire was filled in together with the supervisor, involving a number of fixed choice questions. These, together with the answers, are listed in appendix A.

6.3 Results from the User Study

As a formative evaluation, this study is focused on qualitative aspects, for example how the subjects experience the haptic feedback and how the feedback helps or hinders the process of exploring the data. Both interview answers and feedback through the questionnaire have contributed to the results presented here, but the interviews are the primarily source. This section describes only the opinions and answers from the subjects, which are case specific and associated with the discipline, the subject's previous experience and the interface itself. In the following section a wider discussion is presented where we aim at extracting the more general aspects and conclusions from the results.

During the first phase, using stream-ribbons without haptic feedback, the subjects found that, even though some understanding of the flow could be obtained, it was hard to distinguish between noise and detailed information about the flow. It should be noted that stream-ribbons typically do not distinguish between different vector magnitudes and thus does not allow the user to discriminate between noise and flow data. In particular the direction of the flow was hard to perceive, and so some important information about the heart anatomy was lost. It was also verified that the visual morphological information was insufficient for navigating the anatomy.

In the second phase, using haptics only, some subjects expressed that the local information provided by the haptic feedback made it possible to produce some mental image of the flow distribution. Also, even though the pull from the gradient mode made it easier to find and follow flow and required less precision in the user's movements, some of the subjects expressed that it had a negative impact on the presentation of fine details and that it was therefore not suitable for close examination of the identified flow. The follow mode, both stand-alone and in combination with direct force mapping, was found to be more accurate and conveyed more detail. The direct force mapping was considered to add vital information about flow direction although one subject noted that there was a possibility that it could lead to misinterpretation of the path of the flow. One subject expressed that the haptic feedback was only partly helpful — stating that when the probe is "correctly positioned" the feedback works well, but outside distinct flow it becomes confusing.

With combined haptic and graphical exploration, the third phase, the subjects found the interaction to be considerably improved both with respect to only haptic feedback and only visual feedback. They found the feedback helpful when searching for areas with flow and when trying to distribute stream-ribbons throughout the dataset. While the haptic feedback is the primary means to find and follow flow, the graphical stream-ribbons provide global information and confirmation of the first impression. A majority of the subjects also believed that the presence of haptic feedback helped them understand the distributed flow. The overall opinion was that using the combination gave both faster and more reliable interaction than using either haptics or graphics alone.

This was confirmed in the last phase, when the haptic feedback was again removed. The subjects found that the interaction slowed down and required more concentration. One subject also stated that he found his exploration less structured without the guidance from the haptic feedback. Another subject had expressed that the haptic feedback was hard to understand and believed it would require much more training, but still found that some assistance was lost with the deactivation of the feedback. This subject had bad stereo-vision and expressed that the haptic feedback provided useful guidance in judging depth.

The general opinion on the semi-immersive haptics exploration environment was that it was easy to work with. The haptic feedback was considered helpful and the combination of haptics/graphics produced a better result than using visual feedback alone. We observed that all subjects found controlling the haptic interaction easy to learn and that no-one found the main haptic scheme to have a direct negative impact on the process of exploring and understanding the data.

6.4 Conclusions from the User Study

Potential users have expressed the opinion that deploying combined haptic and visual feedback gives considerable advantages over using a purely graphical interface in the exploration process. This concurs with the results from earlier evaluations of haptic interaction, however the nature of this study leaves the verification of this claim to future, more specialized, studies. A number of more general conclusions, however, can be drawn from the behaviour of our subjects and their comments, regarding both the haptic interaction itself and the design of haptic applications.

Four important aspects of how the haptic feedback may assist the exploration task have been noted:

Physical guidance First of all the feedback can physically guide the user in the exploration process, as anticipated; it can help the user find features by providing physical guides through the volume. For example, the pull from the gradient mode guides the user towards high flow and the follow mode helps the user to find the continuation of an already located path.

Mental guidance The feedback may also, to some degree, help the user to perform a more structured search. Only one of the subjects mentioned this effect. Even so, since this is an abstract notion, we believe that it should be taken into consideration when designing haptic interaction schemes.

Supplementary information The haptic feedback has the potential to convey information that is not represented visually — in this case the direction of the flow.

Complementary information Even if the haptic mode chosen for interaction represents the same features as are visually shown, the feedback can reinforce the visual impression and enhance the understanding. In our example the flow orientation and path is represented by haptic feedback through the follow mode and visually through the interactive stream ribbons. Even so the subjects expressed that the haptic dimension deepened the understanding of the flow data.

The emphasis on these four different aspects varies between modes, and the experience of our test subjects during the exploration depended heavily on the choice of haptic modes and their design. This implies that, for some data, haptics can be of great assistance if the mode is well designed but, if the mode is poorly designed, the haptics can be of little or no help at all or even have a negative impact on the exploration. This has been indicated before[23] and shows the vital importance of tailoring and testing specific haptic schemes for each given problem area and task at hand.

Finding the most appropriate haptic mode or set of haptic modes, with corresponding parameters and transfer functions, to most effectively represent a dataset and facilitate the exploration process is a non-trivial task. The optimal choice of haptic feedback may also differ between different applications, users and tasks. While one user may prefer physical guidance from the haptic instrument, another may feel better assisted by extra information about non-visual properties of the data. This emphasizes the vital importance of tailoring haptic modes and performing iterative development of the application for each given problem area and purpose. There is thus a need to speed up the development process and even allow interactive design of haptic interaction in order to enable users to rapidly:

- choose, try out and replace the haptic modes
- evaluate the impact of the chosen modes
- compare different modes with respect to both the different desired effects of the feedback and the key factors of the current application.

7 Conclusions and Future Work

We have shown how the recently introduced haptic primitives can be used to build “haptic modes”, entities describing the relation between a data type and a haptic representation. With this formalization of volume haptics, the haptic representations of volumetric data can be selected, designed and used in a manner similar to how visual components are setup for a certain problem.

We have presented a toolkit using haptic modes as base components, that aims to remedy the lack of viable solutions for volume haptics. To allow for fast and easy construction, testing and fine-tuning, we have designed our toolkit to form a data processing pipeline, supporting real-time manipulation and configuration. It allows easy construction of a wide variety of haptic schemes and, by careful selection between them, during runtime, various features of the data can be emphasized.

Using the current technology for volume haptics we have performed a formative evaluation identifying four important aspects to consider when designing haptic interfaces for volume data exploration. The study also showed the impact of the form and design of the haptic scheme on its impression and effect. This indicates the importance of interactive design and fine-tuning of haptic interaction schemes to find the optimal balance between the mentioned aspects in a certain task.

These results also indicate that there is great potential for the use of haptics in volumetric data exploration, however there is a need for further studies. For example, there is a need for research on *how* to optimize the design of haptic interaction for an application area, task and user. Furthermore, the connection between data and material properties, here manifested as haptic transfer functions, is not fully understood. Future research include studies on the impact of specific transfer functions on user performance and understanding of the haptic feedback.

Acknowledgements Lars Wigström at the Center for Medical Image Science and Visualization (CMIV) at Linköping University and Mattias Sillén at Saab AB are gratefully acknowledged for providing high quality datasets. The staff at CMIV are also gratefully acknowledged for participation in the pilot study and Lena Tibell at the department of biomedicine and surgery for help with the study.

A The Questionnaire

The questions in the questionnaire are answered with a value between one and five corresponding to “do not agree” and “fully agree”, respectively (Likert scaling). Each question listed below is marked with a letter, corresponding to a line in Fig. 9. The questions have been translated from the original Swedish.

The visual representation...

A ...has a resolution high enough
B ...is well classified
C ...has good update rate

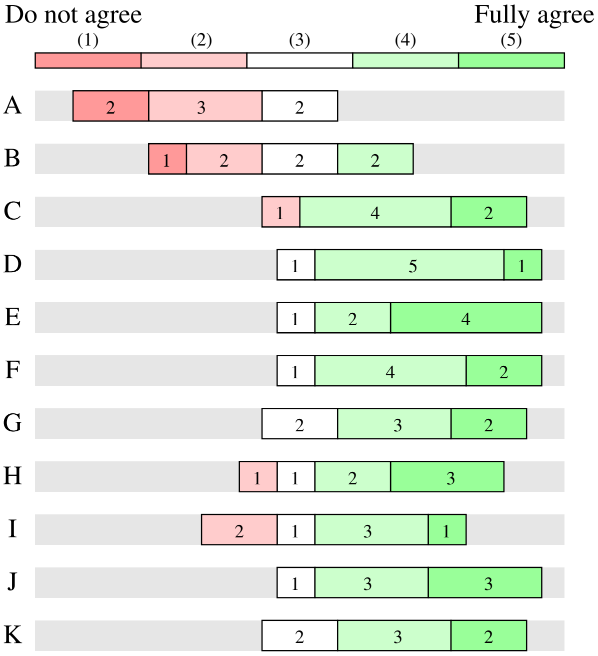


Figure 9 The answers from the questionnaire. Each question is answered using a five level Likert scale. The order and colour of the blocks corresponds to the answer and their size and number corresponds to the number of subjects giving that answer.

The stream-ribbons...

D ...give good visual representation of the flow
E ...are easy to use and handle
F ...behave in a predictable manner

The haptic interaction...

G ...is easy to learn to use
H ...describes the flow in a comprehensive manner
I ...makes it easier to understand the flow
J ...makes it easier to find flow
K ...helps when distributing stream-ribbons

References

1. Steven Wall and William Harwin. Quantification of the effects of haptic feedback during a motor skills task in a simulated environment. In *Proceedings at Phantom User Research Symposium’00*, 2000.
2. Arthur E. Kirkpatrick and Sarah A. Douglas. Application-based evaluation of haptic interfaces. In *Proceedings of the 10th Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 2002.
3. P. J. Passmore, C. F. Nielsen, W. J. Cosh, and A. Darzi. Effects of viewing and orientation on path following in a medical teleoperation environment. In *Proceedings of IEEE Virtual Reality 2001*, 2001.
4. H. Iwata and H. Noma. Volume haptization. In *Proceedings of IEEE 1993 Symposium on Research Frontiers in Virtual Reality*, pages 16–23, October 1993.

5. Richard J. Adams, Daniel Klowden, and Balke Hannaford. Virtual training for manual assembly task. *Haptics-e, The Electronic Journal of Haptics Research (www.haptics-e.org)*, 2(2), October 2001.
6. Steven A. Wall, Karin Paynter, Ann Marie Shillito, Mark Wright, and Silvia Scali. The effect of haptic feedback and stereo graphics in a 3d target acquisition task. In *Proceedings of Eurohaptics*. University of Edinburgh, United Kingdom, 2002.
7. Karljohan Lundin, Björn Gudmundsson, and Anders Ynnerman. General proxy-based haptics for volume visualization. In *Proceedings of the World Haptics Conference*, pages 557–560, Pisa, Italy, March 2005. IEEE.
8. A. Mor, S. Gibson, and J. Samosky. Interacting with 3-dimensional medical data: Haptic feedback for surgical simulation. In *Proceedings of Phantom User Group Workshop'96*, 1996.
9. R. S. Avila and L. M. Sobierajski. A haptic interaction method for volume visualization. In *Proceedings at IEEE Visualization*, pages 197–204, October 1996.
10. W. Hashimoto and H. Iwata. A versatile software platform for visual/haptic environment. *Journal of Control, Automation and Systems Engineering*, pages 106–114, 1997.
11. Farid Infed, Shane V. Brown, Christopher D. Lee, Dale A. Lawrence, Anne M. Dougherty, and Lucy Y. Pao. Combined visual/haptic rendering modes for scientific visualization. In *Proceedings of 8th Annual Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, 1999.
12. Dale A. Lawrence, Christopher D. Lee, Lucy Y. Pao, and Roman Y. Novoselov. Shock and vortex visualization using a combined visual/haptic interface. In *Proceedings of IEEE Conference on Visualization and Computer Graphics*, 2000.
13. Karljohan Lundin, Anders Ynnerman, and Björn Gudmundsson. Proxy-based haptic feedback from volumetric density data. In *Proceedings of Eurohaptics*, pages 104–109. University of Edinburgh, United Kingdom, 2002.
14. Milan Ikits, J. Dean Brederson, Charles D. Hansen, and Christopher R. Johnson. A constraint-based technique for haptic volume exploration. In *Proceedings of IEEE Visualization '03, pp. 263–269*, 2003.
15. Karljohan Lundin, Mattias Sillen, Matthew Cooper, and Anders Ynnerman. Haptic visualization of computational fluid dynamics data using reactive forces. In *Proceedings of Conference on Visualization and Data Analysis, part of IS&T/SPIE Symposium on Electronic Imaging 2005*, pages 31–41, San Jose, CA USA, January 2005.
16. Walter Aviles and John Ranta. Haptic interaction with geoscientific data. In *Proceedings at Phantom User Group Workshop'99*, 1999.
17. Karljohan Lundin, Matthew Cooper, and Anders Ynnerman. The orthogonal constraints problem with the constraint approach to proxy-based volume haptics and a solution. In *Proceedings of SIGRAD Conference*, pages 45–49, Lund, Sweden, November 2005. SIGRAD.
18. Bruce Randall Donald and Frederick Henle. Using haptics vector fields for animation motion control. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2000.
19. Lucy Pao and Dale Lawrence. Synergistic visual/haptic computer interfaces. In *Proceedings of Japan/USA/Vietnam Workshop on Research and Education in Systems, Computation, and Control Engineering*, 1998.
20. Lars Wigström, Lars Sjöqvist, and Bengt Wranne. Temporally resolved 3d phase-contrast imaging. *Magnetic Resonance Medicine*, 36(5), November 1996.
21. Anna Fyrenius, Lars Wigström, Tino Ebbers, Matts Karlsson, Jan Engvall, and Ann F. Bolger. Three dimensional flow in the human left atrium. *Heart*, 86:448–455, October 2001.
22. Lars Wigström, Tino Ebbers, Anna Fyrenius, Matts Karlsson, Jan Engvall, Bengt Wranne, and Ann F. Bolger. Particle trace visualization of intracardiac flow using time-resolved 3d phase contrast mri. *Magnetic Resonance in Medicine*, 41:793–799, 1999.
23. Ross Maciejewski, Seungmoon Choi, David Ebert, and Hong Tan. Multi-modal perceptualization of volumetric data and its application to molecular docking. In *Proceedings of the World Haptics Conference*, pages 511–514, Pisa, Italy, March 2005. IEEE.