22 The Visual Haptic Workbench

MILAN IKITS J. DEAN BREDERSON
Scientific Computing and Imaging Institute, University of Utah

22.1 Introduction

Haptic feedback is a promising interaction modality for a variety of applications. Successful examples include robot teleoperation [57], virtual prototyping [63], painting [4], and surgical planning and training [30,55]. Such applications are augmented with force or tactile feedback for two reasons: (1) To increase the realism of the simulation, and (2) To improve operator performance, which can be measured by precision, fatigue level, and task completion times.

Even though a great variety of graphical visualization techniques have been developed in the past, effective display of complex multidimensional and multifield datasets remains a challenging task. The human visual system is excellent at interpreting 2D images. Understanding volumetric features, however, is difficult because of occlusion, clutter, and lack of spatial cues. Stereoscopic rendering, shadows, and proper illumination provide important depth cues that make feature discrimination easier. Using transparency reduces occlusion and clutter at the price of increasing ambiguity of the visualization.

In contrast to visual displays, haptic interfaces create a tightly coupled information flow via position sensing and force feedback. Such coupled information exchange results in more natural and intuitive interaction and utilizes some of the user's additional sensory-channel bandwidth. When users are presented with a proper combination of visual and haptic information, they experience a sensory synergy resulting from physiological reinforcement of the displayed multimodal cues [19].

Implementations of the traditional visualization pipeline typically provide a limited set of interactive data-exploration capabilities. Tasks such as finding and measuring features in the data or investigating the relationship between different quantities may be easier to perform with more natural data-exploration tools. To develop visualization and exploration techniques that further increase insight and intuitive understanding of scientific datasets, we designed and built an integrated immersive visual and haptic system, the Visual Haptic Workbench (VHW) [10]. In the following sections, we summarize our experiences with this system, discuss relevant issues in the context of developing effective visualization applications for immersive environments, and describe a haptic rendering technique that facilitates intuitive exploration modes for multifield volumetric datasets.

22.2 The Visual Haptic Workbench

The VHW is a testbed system developed primarily for haptic immersive scientific visualization. It is composed of a SensAble PHANToM 3.0L mounted above a Fakespace Immersive Workbench in an inverted configuration (Fig. 22.1). Head, hand, and stylus pose measurements are provided by a Polhemus Fastrak magnetic position tracker. Stereo images are generated by an Electrohome Marquee 9500LC projector and are reflected via folded optics onto the back of the nonlinear diffusion surface of the workbench. A pair of Stereographics CrystalEyes LCD shutter glasses, strobed at a 120 Hz refresh rate, is used for stereo viewing. In a typical scenario, the user's dominant hand manipulates the PHANToM stylus to experience haptic feedback from the virtual scene, and the Q1

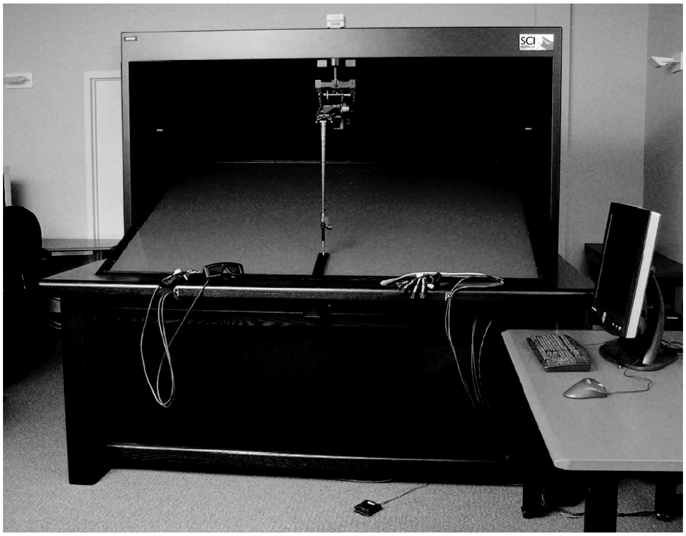


Figure 22.1 The Visual Haptic Workbench integrates a large workspace SensAble PHANToM with a Fakespace Immersive Workbench.

subdominant hand is used for system-control tasks such as navigating a menu interface. A pair of Fakespace Pinch Gloves and a pair of 5DT Data Gloves are provided for implementing more complex interaction techniques. Our custom additions to the workbench hardware include a ‘‘step to operate’’ footswitch instead of the original ‘‘push to interrupt’’ switch, which is used as a more convenient safety mechanism, a registration apparatus for placing the PHANToM in a fixed position on the surface of the workbench during encoder initialization, and an inexpensive 6DOF interaction device, the I^3Stick [9]. The system is constructed in such a way that it can be connected to a PC with the latest available graphics card without further modifications.

Compared to other similar systems, e.g., the UNC nano Workbench [21] or the CSIRO Haptic Workbench [68], our setup has the advantage of facilitating whole-arm interaction, using a wide-angle head-tracked visual display, and providing direct (1:1) correspondence between the visual and haptic workspaces. We found that tilting the workbench surface at a 20° angle both increases the visual range and aligns the hotspots of the workspaces [46]. Placing the PHANToM arm in front of the projection screen has the disadvantage of occludin the view, reducing the size of the available stereoscopic workspace. A related problem is that the low stiffness of the arm is apparent during hard surface contact, since the PHANToM end-effector may visually penetrate the virtual surface. To reduce these problems, we use a fixed offset between the actual endpoint and its virtual representation. Mounting the haptic device behind the screen would also solve these problems. Unfortunately, this is possible with front-projection screens only. Using front projection, however, further reduces the size of the visual workspace, because the projection screen has to be located closer to the eyes than in other cases.

There are several important issues to consider when developing visualization applications for immersive environments. In the following subsections we discuss some of our observations and summarize what we have learned from our experiences with the Visual Haptic Workbench.

22.2.1 Calibration and Registration

Many immersive virtual-reality applications benefit from precisely calibrated system components. Greater accuracy is desired though, to provide users with a more compelling experience and to increase precision and reduce frustration during 6DOF interaction tasks. Experimental test-bed systems require very accurate registration; otherwise, registration artifacts may be difficult to separate from other experimental factors [70]. Visual artifacts caused by registration errors include misalignment of real and virtual objects, as well as the notorious "swimming" effect, i.e., the motion and changing shape of stationary objects as the user moves around in the environment. It is also important to make sure that the generated visual and haptic cues match by precisely colocating the various workspaces of the system.

Registration error sources can be categorized according to whether they produce *geometric* or *optical* distortions. Geometric errors are the result of inaccurate tracking, system delay, misalignment of coordinate systems, and imprecise viewing and interaction parameters. Optical errors, caused by the limitations of the image-generation subsystem, are manifested as convergence and aliasing problems, display nonlinearities, and color aberrations. Haptic-rendering fidelity largely depends on the structural and dynamic characteristics of the haptic interface, the accuracy of its kinematic description, the update rate, and the control algorithm used to produce reaction forces and torques.

Possible geometric error sources for the VHW include tracker distortion and the unknown rigid-body transformations between the coordinate frames attached to the tracker transmitter, the PHANToM base, the display surface, and the eyepoints and interaction-device hotspots relative to the tracker sensors. Ideally, we want to reduce the discrepancies in these parameters so that the overall registration error does not exceed a few millimeters. In previous work, an external measurement device was used for coregistering the components of the nano Workbench [21]. We have experimented with calibrating the magnetic tracker of our system using an optical device and found that it is possible to reduce measurement errors to a few millimeters within a large portion of the workspace [27]. We have also developed and evaluated a method for rapidly calibrating and registering the system components without using external metrology [26,29]. Our results indicate that to reach the desired level of accuracy, we need to replace the magnetic tracker with an accurate, low-latency optical solution.

22.2.2 Interaction Techniques

One of the "grand challenges" of using immersive environments for scientific exploration is "making interaction comfortable, fast, and effective" [72]. Designing and evaluating interaction techniques and user interfaces is an important area of virtual environment research [7]. Even though immersive virtual reality provides the possibility for more natural interaction, working with a computer-generated 3D world is difficult, because the haptic cues that are part of the real world are missing from these environments. In the past, a variety of complex interaction techniques that are not particularly applicable for everyday use have been developed for immersive environments. In contrast, the desktop WIMP (Windows, Icons, Menus, Pointers) paradigm has been very successful due to its simplicity, robustness, and convenience. We found that the following guidelines should be observed when developing interaction techniques for immersive visualization applications [64]:

- Avoid complex and cumbersome devices, e.g., gloves.
- Use intuitive and simple interaction metaphors; reserve "magic" techniques for expert use, e.g., for shortcuts or text input [22,8].
- Utilize two-handed manipulation when possible, but provide ways to perform the same task with a single hand.

- Use physical and virtual constraints to increase precision and reduce fatigue.

Q2

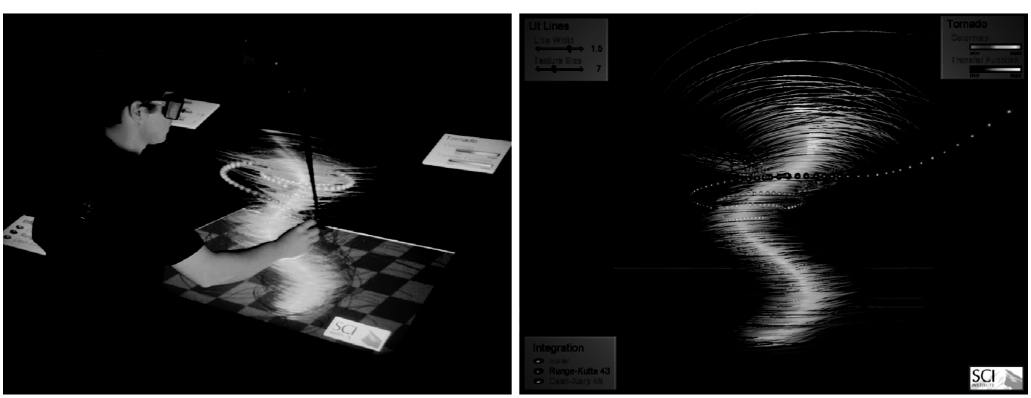
Direct manipulation widgets provide a convenient means for exploring 3D datasets in desktop applications [15]. Typically, 3D widgets are implemented using the OpenGL picking and selection mechanism, or supported by the scene-graph API upon which the application is built [69,75]. Widgets are at least as useful for immersive visualization, but the lack of physical constraints can make them cumbersome to use. We have developed a library of interface components for building applications that run in both desktop and immersive environments without modification (Fig. 22.2). Adding haptic feedback to the interface components is an area of future research.

22.2.3 Software Framework

Creating successful immersive applications is inherently an interative process. An active area of virtual-environment research is making application development comfortable, fast, and effective. Application development and evaluation, however, usually happen in two different workspaces. Ideally, the developer should be able to run an application on every available platform without modifications, using an interface optimized to that particular platform [33] (Fig. 22.2). Previous efforts to create an immersive tool that could also run on the desktop required significant contributions from the programmer, because the interface remained platform-dependent [54]. Currently, most frameworks do not support this concept, and provide a *simulator mode* instead [37,5,16]. In this mode, a third-person view of the user is presented in such a way that immersive controls are mapped to a 2D desktop interface. Even though this mode is useful for examining how the user's actions effect the environment and vice versa, it prevents the developer from focusing on the content of the application.

22.2.4 Visualization Methods

Desktop visualization applications are event-based, and the majority of the events originate from the user. The viewer is in a fixed position, so the update rate and latency of interaction are less critical. In contrast, virtual-environment applications are built upon a continuous simulation with stringent requirements, including high update rate and low system latency, similar to those of computer games [12]. Thus, visualization techniques have to strike a balance between achievable quality and rendering



(a) (b)

Figure 22.2 (a) A user explores a tornado dataset on the Visual Haptic Workbench. (b) Screenshot of the same application running in a desktop environment. Dataset courtesy of R. Crawfis, Ohio State University, and N. Max, Visualization Laboratory, Lawrence Livermore National Laboratory.

speed. Adaptations of traditional visualization methods have relied on multiresolution representations to maintain fluid interaction between the user and the application [13,20,71].

22.3 Haptic Rendering Techniques for Scientific Visualization

The majority of haptic rendering algorithms are geometric in nature, since they deal with the problem of interacting with various surface representations at real-time update rates. Surface rendering requires a suitable geometric model, typically combined with a bounding volume hierarchy, a rapid collision-detection technique, an incremental surface-tracing algorithm, and a model for generating contact forces from the probe-surface interaction. Surface-tracing algorithms exist for a variety of representations, including polygonal, parametric Q3 (NURBS), and implicit surfaces. These algorithms rely on a combination of global and local distance queries to track the geometry closest to the interaction point. Haptic surface rendering has evolved from simple force-field methods [43] to constraint-based approaches that utilize a proxy point [77,58]. More recently, efficient techniques have emerged for haptic display of contact between complex polygonal objects [34,51].

Contact forces are usually modeled by interactions between the probe and a rigid viscoelastic surface. A virtual spring and damper are used to mechanically couple the probe with the proxy during contact. From the visualization point of view, surfaces are represented by a set of unilateral constraints that prevent the proxy from penetrating the object. Previous research has focused on improving the perceived crispness of surfaces, and on augmenting them with various material properties to create realistic and convincing virtual objects [61,44,67,59,40].

Early work in haptic visualization used simple volumetric methods for exploring scalar and vector fields as well as molecular interactions [11,32]. The majority of previous methods for haptic display of volume data properties are based on a functional relationship between the reflected force and torque vectors, and the probe state and local data measures:

$$\vec{F} = \vec{F}(X,\ D,\ T) \tag{22.1}$$

where X denotes the state, typically position $\vec{x}$ and velocity $\dot{\vec{x}}$ of the haptic probe, D represents a set of local data measures at the probe position, and T stands for a set of haptic transfer functions and rendering parameters. We borrow the term *force-field rendering* for this class of techniques. The simplest examples in this category include density-modulated viscous drag for scalar data [3,52] and direct display of vector data [32,42]:

$$\vec{F}(\{\vec{x},\ \dot{\vec{x}}\},\ \{s(\vec{x})\},\ \{k(s)\}) = -k(s(\vec{x}))\dot{\vec{x}} \tag{22.2}$$

$$\vec{F}(\{\vec{x}\},\ \{\vec{v}(\vec{x})\},\ \{k\}) = k\vec{v}(\vec{x}) \tag{22.3}$$

where the gain k is adjusted according to the scale and magnitude of the data measures and the capabilities of the haptic interface. Note that in Equation 22.2 we modulate viscous drag as a function of data value and in Equation 22.3 we apply a force directly proportional to the local field vector.

Even though this approach represents an important step in the evolution of haptic data-rendering techniques, it suffers from several limitations. First, it provides limited expressive power, because it is difficult to display and emphasize features in a purely functional form. For example, we found that using complex transfer functions for rendering isosurfaces is less convincing than traditional surface-rendering approaches [3,31,39]. The reason for this is that the notion of *memory* is missing from these formulations [60,41]. Second, the device capabilities are captured implicitly in the rendering parameters. Applying a force as a function of the probe state can easily result in instability, especially when several rendering modes are combined. In general, it is very difficult and tedious to tune the behavior of the dynamic system formed by the force-field equation (Equation 22.1) and the motion equations of the haptic device by finding an appropriate set of rendering parameters.

Fortunately, haptic-rendering stability can be guaranteed with use a virtual coupling network [14,1]. The coupler acts as a low-pass filter between the haptic display and the virtual environment, limiting the maximum impedance that needs to be exhibited by the device and preventing the accumulation of energy in the system [56,24]. Although the coupler is not part of the environment, the commonly used spring-damper form had been introduced implicitly in constraint-based surface-rendering algorithms. In the next section, we describe a similar approach to haptic rendering of directional information in volumetric datasets.

22.4 Data Exploration with Haptic Constraints

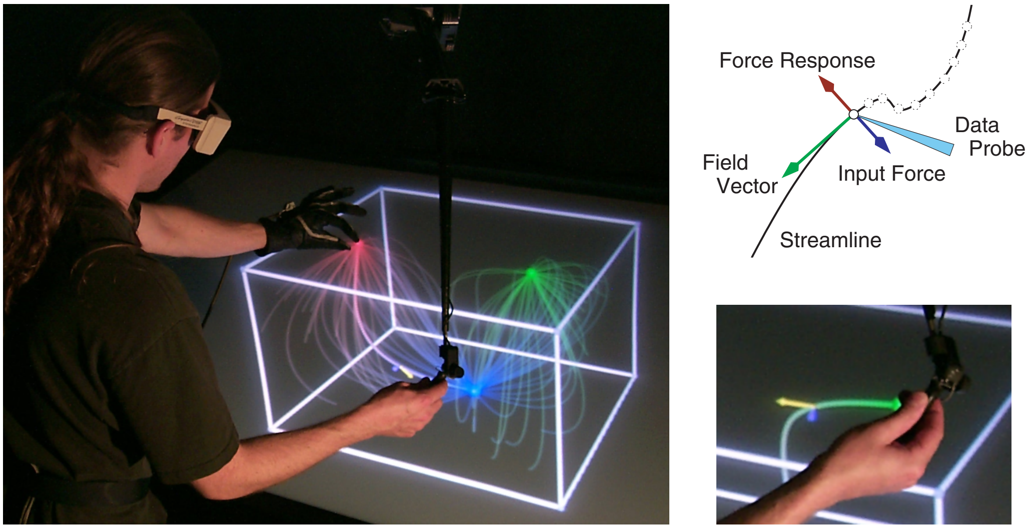
Constraints have been used successfully in both telerobotics and haptics applications. In early work, virtual fixtures or guides improved operator performance in robot teleoperation tasks [57]. More recently, a haptic-rendering framework was developed with algebraic constraints as the foundation [25]. Haptic constraints have helped guide users in a goal-directed task [23]. User interfaces can also benefit from guidance. Examples include a haptic version of the common desktop metaphor [47] and a more natural paradigm for media control [66].

We found that constraints provide a useful and general foundation for developing haptic-rendering algorithms for scientific datasets [28]. For example, constrained spatial probing for seeding visualization algorithms local to the proxy, e.g., particle advection, typically results in more cohesive insight than its unconstrained version. Volumetric constraints are obtained by augmentation of the proxy with a local reference frame, and control of its motion according to a set of rules and transfer functions along the axes of the frame. This approach has the advantage of providing a uniform basis for rendering a variety of data modalities. Thus, similar or closely related methods can be applied to seemingly unrelated datasets in such a way that the result is a consistent interaction experience. For example, to guide the user in vector-field data, the proxy can be constrained along a streamline such that any effort to move the probe in a direction perpendicular to the current orientation of the field results in a strong opposing force (Fig. 21.3b). However, if the user pushes the probe hard enough, the proxy could "pop over" to an adjacent streamline, allowing the user to move the probe in three dimensions and still receive strong haptic cues about the orientation of the flow. We can use an additional force component along the streamline to indicate the magnitude of the field. Alternatively, a secondary constraint can be added to convey information about the speed of the flow in the form of haptic "tickmarks." We found that such techniques result in intuitive feedback in exploration of vector-field data. A recent study on the effectiveness of various haptic rendering techniques for CFD datasets reached a similar conclusion [73].

Algorithms for constrained point-based 3DOF haptic rendering have recently been developed for scalar density data [6,41] as well as vector fields used in computational fluid dynamics (CFD) visualization and animation motion-control applications [18,73]. Haptic constraints have also been successfully used for displaying molecular flexibility [38]. Applications that require complex proxy geometry transform the proxy to a point shell to perform approximate 6DOF force and torque calculations using the individual point locations [45,56,42,53]. In recent work, a spherical approximation of tool–voxel interaction was used to speed up intersection calculations in a bone-dissection task [2].

22.4.1 Haptic Rendering with a Constrained Proxy Point

In general, haptic volume-rendering algorithms based on a proxy point include four components



(a) (b)

Figure 22.3 (a) A user explores a volumetric vector dataset. (b) The data probe is constrained along a streamline, resulting in intuitive haptic feedback.

Q4

that are executed at every iteration of the servo loop (Fig. 22.4):

1. **Compute local data measures at current proxy location**: Data values and other measures, e.g., gradient or curvature information, are obtained from interpolation of data elements around the current proxy location. Typical methods include linear and tri-linear interpolation, although higher-order techniques may be more appropriate depending on the scale and resolution of the display [62]. Since haptic rendering is a local process, like particle advection, point-location algorithms for vector-field visualization on curvilinear and unstructured grids are readily applied [49]. A *local reference frame* $(\vec{e}_1, \vec{e}_2, \vec{e}_3)$ is a key component of constraint-based techniques. Examples include the frame defined by the gradient and principal curvature directions in scalar data and the frame of eigenvectors in diffusion-tensor data. Note that the reference frame may be ill-defined or may not exist. Thus, an important

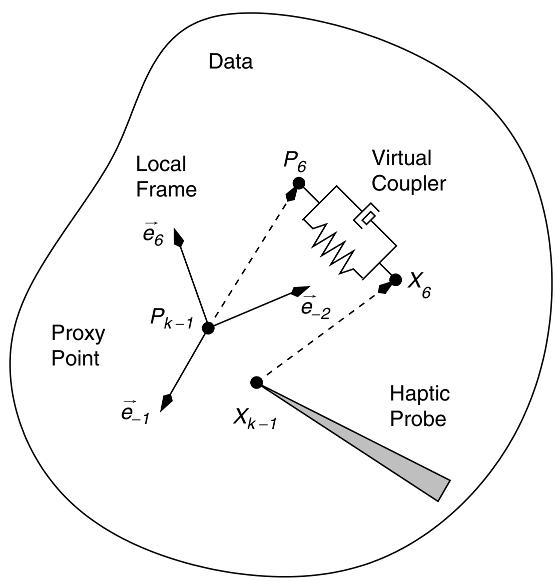


Figure 22.4 Components of constrained point-based 3DOF haptic data rendering. At time-step k, the state of the haptic probe has changed from X_{k-1} to X_k. The proxy state gets updated from P_{k-1} to P_k, from which the force response is computed using a virtual coupler. The p-Proxy update is based on data measures at the previous proxy location, as well as haptic transfer functions and rendering parameters.

requirement for the algorithm is to compute a stable force response even when transitioning into and out of homogeneous and ill-posed regions in the data. For example, in scalar volumes the reference frame is poorly defined in regions where the gradient vanishes. One way to achieve smooth transitioning is to modulate the force output as a function of gradient magnitude [41]. Another example is specifying transfer functions such that isotropic regions are handled properly in diffusion-tensor data. In this case, the transfer function has to be constructed in such a way that the force output either vanishes or degenerates to an isotropic point constraint.

2. **Evaluate haptic transfer functions to determine rendering parameters**: Similar to that of graphical visualizations, the goal of haptic transfer functions is to emphasize and combine features in the data. For example, a transfer function can be used to specify apparent stiffness and friction for isosurface regions based on data value and gradient magnitude [41]. In contrast to visual transfer functions, the design of haptic transfer functions is an unexplored area. Although it is possible to faithfully reproduce measured material properties [50], synthesizing them from different or abstract data remains difficult. In the examples presented in Section 22.5 we utilize stiffness and drag threshold transfer functions $\vec{k}$ and $\vec{\tau}$ to constrain the motion of the proxy along the axes of the local reference frame.

3. **Update proxy state**: In this step, the state of the proxy is updated according to simple motion rules. We have chosen a purely geometric approach that updates the proxy location based on probe motion and rendering parameters along the axes of the local frame:

$$\vec{p}_k = \vec{p}_{k-1} + \Delta\vec{p} = \vec{p}_{k-1} + \sum_{i=1}^{3} \Delta p_i \, \vec{e}_i \qquad (22.4)$$

where Δp_i is a function of probe position relative to the previous proxy location, $\Delta x_i = (\vec{x}_k - \vec{p}_{k-1}) \cdot \vec{e}_i$. For example, surface-haptics algorithms locally constrain the proxy to the tangent plane by setting the normal component of change to zero. More sophisticated strategies incorporate the force response from previous steps as well as other stated variables. For example, physically based models assume the proxy has mass m and is moving in a medium with viscosity b [65]: Q5

$$m\ddot{p}_i + b\,\dot{p}_i = F_i \qquad (22.5)$$

where F_i is the force component acting on the proxy point along $\vec{e}_i$. Friction effects can be incorporated by addition and moving of a static friction point within the constraint subspace [61].

Note that the linear approximation used in Equation 22.4 is not always appropriate for expressing a nonlinear constraint, such as staying on a surface or following a streamline. For example, when tracing volumetric isosurfaces, the first-order approximation obtained by projecting the probe point to the tangent plane defined by the gradient at the proxy location will result in the algorithm's quickly losing track of the surface. Thus, we find the new proxy location $\vec{p}_k$ by refining the initial estimate using Newton-Raphson iteration along the gradient direction [60]:

$$\Delta\vec{p} = -\frac{(s(\vec{p}) - s_0)\nabla s\,(\vec{p})}{|\nabla s\,(\vec{p})|^2} \qquad (22.6)$$

where s_0 is the target iso-value. The refinement is terminated when the step size $|\Delta\vec{p}|$ either is sufficiently small or reaches the maximum number of iterations permitted. Similarly, higher-order integration schemes, e.g., the fourth-order Runge-Kutta method, are necessary for computing the reference direction when following streamlines in vector data. For larger step sizes, supersampling and iteration of steps 1–3 may be required to ensure that constraints are satisfied accurately [60,6].

Linearized constraints can be applied in arbitrary order if the reference frame is orthogonal. For nonlinear constraints and nonorthogonal reference frames, the order of application defines which constraint is considered primary, which is considered secondary, *etc*. For example, to follow streamlines on a surface, we first move the proxy along the local field direction, then project it to the tangent plane of the surface. If the vector field has out-of-plane components, this order of steps corresponds to projecting the vector field onto the tangent surface. Reversing the order results in a different proxy location and creates a different haptic effect.

22.4.1.4 Compute force response:

When using the spring-damper form of virtual coupling, the force response is computed from

$$\vec{F}_k = k_c\ (\vec{x}_k - \vec{p}_k) - b_c\ (\dot{\vec{x}}_k - \dot{\vec{p}}_k) \tag{22.7}$$

where k_c and b_c are chosen according to the device capabilities. The optimal choice maximizes the coupling stiffness without causing instability [1]. One problem is that these parameters may not be constant throughout the workspace. A choice that works well in the center may cause instability near the perimeter. Nevertheless, we can tune them by applying a point constraint at different locations in the workspace and determining which settings cause the device to become unstable on its own, i.e., without a stabilizing grasp. Analysis of the parameters could reveal the optimal operational region within the workspace of the device. In our implementation, we exclude the second term from Equation 22.7, since filtering velocity is difficult without high-resolution position measurements [14].

22.4.2 Motion Rules and Transfer Functions

Motion rules allow us to create various haptic effects that we can further modulate via haptic transfer functions. One effect simulates plastic material behavior by generating increasing resistance between the probe and the proxy until a certain threshold is reached. At this point, the proxy is allowed to move towards the probe, keeping the reaction force at the same level. This effect is expressed succinctly by the following formula: Q6

$$\Delta p_i = \mathrm{sgn}(\Delta x_i) \max(|\Delta x_i| - \tau_i,\ 0) \tag{22.8}$$

This model yields free-space motion when $\tau_i = 0$:

$$\Delta p_i = \Delta x_i \tag{22.9}$$

and a *bilateral* constraint when $\tau_i > 0$. We use the term *drag threshold* for τ_i, because it controls the difficulty of dragging the proxy along axis $\vec{e}_i$. Note that a stationary constraint is obtained when τ_i is sufficiently large, because it would take considerable effort to move the probe away from the proxy while resisting the increasing amount of force between them.

A *unilateral* constraint, which is the basis for surface-rendering algorithms, is obtained by considering the direction of travel along the axis:

$$\Delta p_i = \begin{cases} \Delta x_i & \text{if } \Delta x_i > 0 \\ \max(\Delta x_i - \tau_i,\ 0) & \text{if } \Delta x_i \le 0 \end{cases} \tag{22.10}$$

A bilateral *snap-drag* constraint changes the proxy location in discrete steps:

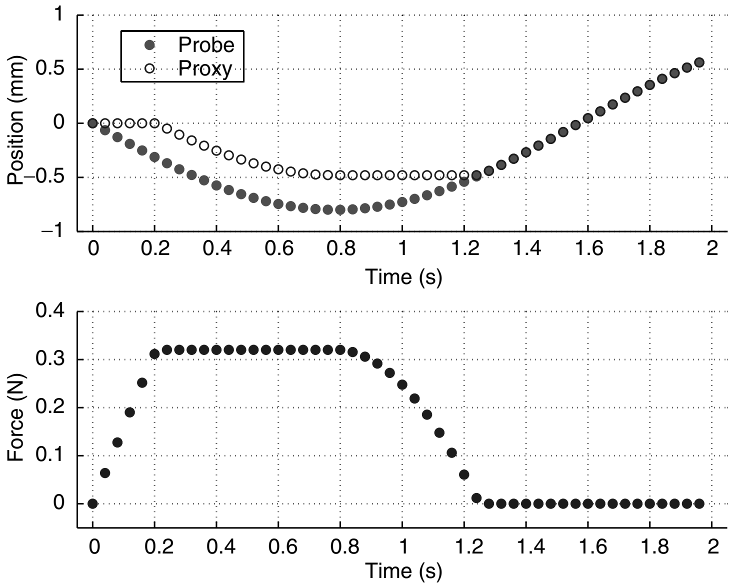
$$\Delta p_i = \begin{cases} \tau_i & \text{if } \Delta x_i > \tau_i \\ 0 & \text{if } \Delta x_i \le \tau_i \end{cases} \tag{22.11}$$

The latter two rules are shown in Fig. 22.5, along with the resulting force responses.

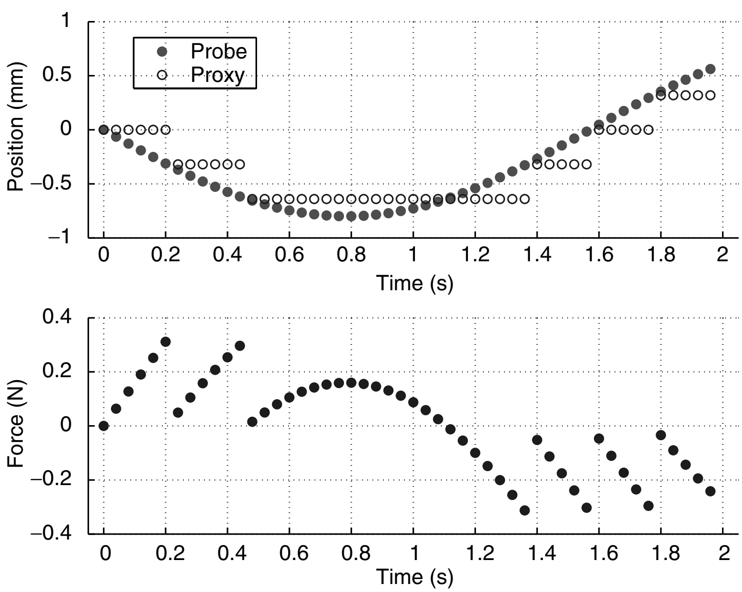
We can influence proxy motion indirectly by scaling the force output according to stiffness transfer function $\vec{k}$:

$$F_{k,i} = k_i k_c\ (x_{k,i} - p_{k,i}) \tag{22.12}$$

where $0 \le k_i \le 1$. This reduces the force required for dragging the proxy. Note that setting either τ_i or k_i to zero produces no force output and creates frictionless motion along the axis. However, it yields two different proxy behaviors, since in the first case the proxy follows the



(a) Unilateral constraint.



(b) Snap-drag constraint.

Figure 22.5 Examples of the 1D motion rule: (a) unilateral drag and (b) bilateral snap-drag. The motion of the probe and the proxy as a function of time is represented by the filled and empty circles, respectively. The resulting force responses are shown in the lower parts of the figures. Note that the sampling does not correspond to the haptic update rate.

probe exactly, while in the second case it lags behind by distance τ_i. Both parameters are necessary, because we want to express a range of effects, from subtle directional hints to stiff rigid constraints, in addition to both elastic and plastic material behavior.

22.5 Examples

In the following subsections, we describe how haptic constraints aid the user in two exploratory tasks: (1) Investigating the relationship between cardiac muscle fibers and potential distributions, and (2) Exploring the connectivity of brain white matter in diffusion-tensor MRI data.

22.5.1 Tracing Heart Muscle Fibers

Particle advection, i.e., integrating the motion of massless particles with velocities defined by the field, is a basic building block of vector-and tensor-field visualization techniques. The haptic equivalent is achieved by restriction of the motion of the proxy along the path of a single particle (Fig. 22.3a).

This method is easily modified to display orientation information on isosurfaces. Such a technique could be useful for investigating the relationship between heart muscle fiber orientations and potential distributions resulting from cardiac bioelectric finite-element simulations [48]. These simulations are typically carried out on a curvilinear grid that forms a number of epicardial and endocardial layers (Fig. 22.6). In our implementation, we reorganize the data to an unstructured tetrahedral grid by computing a Delaunay triangulation of the original data points. We assign a scalar value to the nodes in individual layers, in increasing order from inside to outside, such that isosurfaces of this scalar field correspond to muscle layers in the model. The gradient field computed from a central difference–approximation formula is used in the iterative refinement Equation 22.6 to make sure the proxy stays on the currently selected layer.

To avoid singularities when interpolating fiber orientation vectors within a tetrahedral element, we use component-wise linear interpolation of the tensor field, obtained by taking the outer product of the vectors with themselves. The major eigenvector of the interpolated tensor yields a smooth orientation field within a tetrahedral element, even when the vectors at the nodes point in completely different directions.

In this example, a local reference frame is formed by the interpolated fiber-orientation and gradient vectors. The snap-drag motion rule allows the user to explore a single layer and "pop through" to a neighboring layer by pushing against the surface. In this case, the drag threshold τ_i is not used for moving the proxy after breaking away from the current surface. Instead, we detect when the probe crosses a neighboring layer and set the proxy location to a numerical approximation of the intersection point. A secondary snap-drag rule constrains proxy motion along the fibers on the surface, allowing the user to switch to a nearby streamline in discrete steps. This method essentially creates a haptic texture on the surface composed of tiny valleys and ridges corresponding to the muscle fibers. See Fig. 22.6 for an illustration of this example.

22.5.2 Exploring Diffusion-Tensor Fields

Diffusion-tensor fields are difficult to comprehend because of the increased dimensionality of the data values and complexity of the features involved. Direct methods, such as glyphs and reaction-diffusion textures, work well on 2D slices, but they are less successful for creating 3D visualizations. Intermediate representations created by adaptations of vector-field techniques result in intuitive visual representations, but fail to capture every aspect of the field [17,74]. Interactive exploration has helped users interpret the complex geometric models that represent features in the data [76]. Our goal is to aid the exploration process by adding

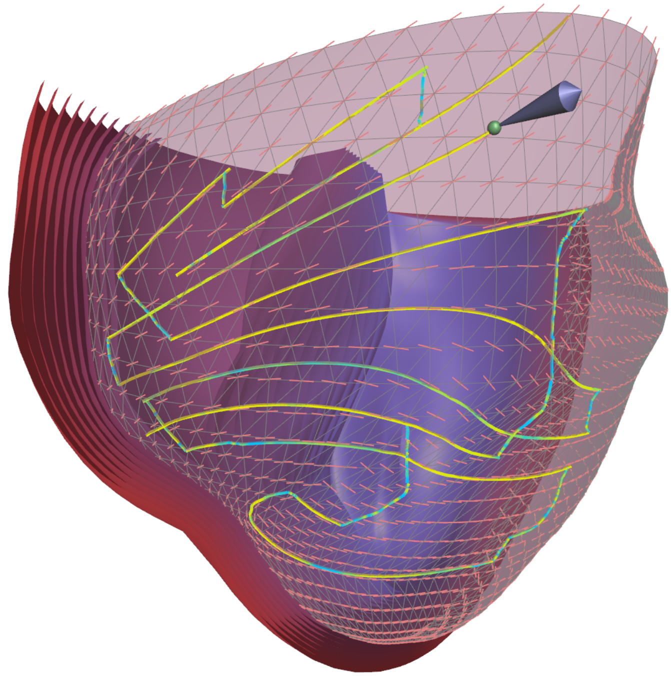


Figure 22.6 Exploring epicardial muscle fibers with haptic feedback. The probe is constrained to follow the local fiber orientation on the surface of a single layer. The user can "pop through" to a neighboring layer by pushing against the surface. Similarly, the user can choose a different fiber by pushing perpendicular to the currently selected fiber while staying on the surface. This effect feels as if the surface were textured with tiny valleys and ridges. The image shows the path of the proxy colored according to the magnitude of the applied force component perpendicular to the fiber orientation and tangent to the surface, from yellow to cyan, indicating increasing tension between the probe and the proxy. The dataset consists of about 30,000 nodes and 200,000 tetrahedral elements. Dataset courtesy of P. Hunter, Bioengineering Institute, University of Auckland.

haptic feedback that guides the user according to the local orientation and anisotropy of the field.

The rate and directionality of water diffusion in tissues is indicated by a second-order symmetric tensor. Anisotropy of the diffusion process can be characterized by the following barycentric measures [36]:

$$c_l = \frac{\lambda_1 - \lambda_2}{\lambda_1 + \lambda_2 + \lambda_3} \tag{22.13}$$

$$c_p = \frac{2(\lambda_2 - \lambda_3)}{\lambda_1 + \lambda_2 + \lambda_3} \tag{22.14}$$

$$c_s = \frac{3\lambda_3}{\lambda_1 + \lambda_2 + \lambda_3} = 1 - c_l - c_p \tag{22.15}$$

where $\lambda_1 \geq \lambda_2 \geq \lambda_3$ are the sorted eigenvalues of the diffusion-tensor matrix. These measures indicate the degree of linear, planar, and spherical anisotropy, respectively. The associated eigenvectors $\vec{e}_1$, $\vec{e}_2$, $\vec{e}_3$ form an orthonormal frame corresponding to the directionality of diffusion. Regions with linear and planar anisotropy represent important features in the data, such as white-matter fiber bundles in brain tissue.

One way to use haptic feedback to indicate tensor orientation and degree of anisotropy is to control proxy motion such that it moves freely along the direction of the major eigenvector but is constrained in the other two directions. We found that setting the drag thresholds to a function of the anisotropy measures results in the desired feedback:

$$\tau_1 = 0 \tag{22.16}$$

$$\tau_2 = \tau(c_l) \tag{22.17}$$

$$\tau_3 = \tau(c_l + c_p) \tag{22.18}$$

where $\tau(x)$ is a monotonically increasing function on $[0 \ldots 1]$. This choice ensures that the transfer functions yield a line constraint along the major eigenvector in regions with linear anisotropy ($c_l \gg c_p,\ c_s$), yield a plane constraint in regions with planar anisotropy ($c_p \gg c_l,\ c_s$), and allow free motion along all three directions in isotropic areas ($c_s \gg c_p,\ c_l$). Recall that the three indices sum to one, so when any one index dominates, the transfer functions emphasize the corresponding type of anisotropy. Alternatively, we can set the threshold to a constant value for all three directions and vary the stiffness similarly to Equation 22.18. In our implementation, we chose a linear ramp for $\tau(x)$, but other possibilities may be more appropriate.

The technique is illustrated in Fig. 22.7. We have observed that it takes little effort to trace out curves indicating fiber distribution and connectivity. Note that numerical methods for fiber tractography require careful specification of initial and stopping conditions, and cannot be used straightforwardly for investigation of connectivity in regions of the data.

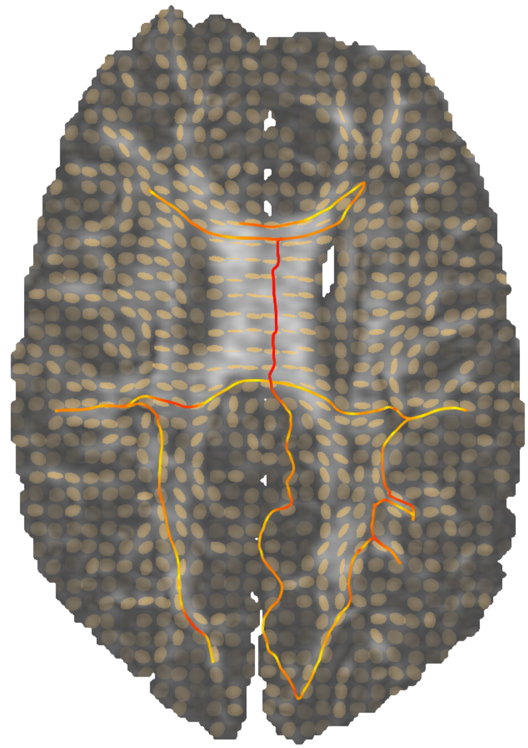


Figure 22.7 Exploring a 148 × 190 DT-MRI slice with haptic feedback. The ellipses represent local diffusion anisotropy and orientation. Lighter areas have higher associated anisotropy. The proxy path is colored according to the magnitude of the applied force, from yellow to red, indicating a larger tension between the probe and the proxy. The curves are tangent to the direction of the major eigenvector of the diffusion-tensor matrix in anisotropic areas. Dataset courtesy of G. Kindlmann and A. Alexander, W. M. Keck Laboratory for Functional Brain Imaging and Behavior, University of Wisconsin-Madison.

22.6 Summary and Future Work

We have designed and built a prototype system for synergistic display of scientific data. By developing and demonstrating initial applications, we have been able to refine our system and identify several important research issues in the context of building effective visualization applications for immersive environments. In the future, we plan to extend our collection of visualization techniques for the exploration of a variety of multidimensional and multifield datasets.

The presented approach for haptic data exploration has several desirable properties: it provides a unified rendering framework for different data modalities, allow secondary effects such as texture and friction to be easily realized, makes haptic transfer functions intrinsic to the algorithm, and allow control parameters to be be tuned to the operational characteristics of the interface device.

A particular challenge we intend to address in the future is the issue of synthesizing useful haptic transfer functions from the underlying data. Investigating the synergistic relationship between visual and haptic transfer functions is another interesting research topic. A disadvantage of using the spring-damper form of virtual coupling is that it is too conservative, meaning that it may limit the efficient display of subtle haptic effects. We will experiment with a recent energy-based approach that uses a time-domain passivity observer and controller to adaptively adjust the coupling parameters [24]. In addition, we plan to extend the haptic rendering method to 6DOF devices. Transforming the constraint-based approach to a purely functional formulation would provide a very natural space for specifying rendering parameters. Finally, the real challenge for synergistic data display is validation. We intend to quantify the usability of our techniques and identify specific combinations that are useful to scientists who directly benefit from synergistic display of their datasets.

Acknowledgments

The authors are grateful to Gordon Kindlmann for enlightening and refreshing discussions as well as the Teem toolkit [35], which allowed efficient dataset probing and manipulation in this work. We thank Robert MacLeod for suggesting the heart visualization example and Hope M.F. Eksten for her help with various construction and fabrication projects. Support for this research was provided by NSF grant ACI-9978063, ARO DURIP grant DAAG-559710065, and the DOE Advanced Visualization Technology Center (AVTC).

References

1. R. J. Adams and B. Hannaford. A two-port framework for the design of unconditionally stable haptic interfaces. In *Proc. IEEE International Conference on Intelligent Robots and Systems*, pages 1254–1259, Victoria, BC, 1998.
2. M. Agus, A. Giachetti, E. Gobetti, G. Zanetti, and A. Zorcolo. Real-time haptic and visual simulation of bone dissection. *Presence: Teleoperators and Virtual Environments*, 12(1): 110–122, 2003.
3. R. S. Avila and L. M. Sobierajski. A haptic interaction method for volume visualization. In *Proc. IEEE Visualization*, pages 197–204, San Francisco, CA, 1996.
4. W. Baxter, V. Scheib, M. C. Lin, and D., Manocha. DAB: Interactive haptic painting with 3D virtual brushes. In *Proc. ACM SIGGRAPH*, pages 461–468, Los Angeles, CA, 2001.
5. A. Bierbaum, C. Just, P. Hartling, K. Meinert, A. Baker and C. Cruz-Neira. VR Juggler: A virtual platform for virtual reality application development. In *Proc. IEEE Virtual Reality*, pages 89–96, Yokohama, Japan, 2001.
6. D. J. Blezek and R. A. Robb. Haptic rendering of isosurfaces directly from medical images. In *Proc. Medicine Meets Virtual Reality*, pages 67–73, San Francisco, CA, 1999.
7. D. A. Bowman, E. Kruijff, J. J. LaViola, Jr., and I. Poupyrev. An introduction to 3D user interface design. *Presence: Teleoperators and Virtual Environments*, 10(1):96–108, 2001.
8. D. A. Bowman, C. J. Rhoton, and M. S. Pinho. Text input techniques for immersive virtual environments: an empirical comparison. In *Proc. Human Factors and Ergonomics Society Annual Meeting*, pages 2154–2158, 2002.
9. J. D. Brederson. The I^3 Stick: An inexpensive, immersive interaction device. *Technical Report UUCS-99–016*, School of Computing, University of Utah, 1999.
10. J. D. Brederson, M. Ikits, C. R. Johnson, and C. D. Hansen. The visual haptic workbench. In *Proc. PHANToM Users Group Workshop*, Aspen, CO, 2000.
11. F. P. Brooks, M. Ouh-Young, J. J. Batter, and P. J. Kilpatrick. Project GROPE – haptic displays for scientific visualization. In *Proc. ACM SIGGRAPH*, pages 177–185, Dallas, TX, 1990.
12. S. Bryson. *Virtual Reality Applications*, pages 3–15. Burlington, MA, Academic Press, 1995.
13. P. Cignoni, C. Montani, and R. Scopigno. MagicSphere: an insight tool for 3D data visualization. *Computer Graphics Forum*, 13(3): 317–328, 1994.
14. J. E. Colgate and J. M. Brown. Issues in the haptic display of tool use. In *Proc. IEEE International Conference on Intelligent Robots and Systems*, pages 140–145, Pittsburgh, PA, 1995.
15. D. B. Conner, S. S. Snibbe, K. P. Herndon, D. C. Robbins, R. C. Zeleznik, and A. van Dam. Three-Dimensional widgets. In *Proc. ACM Symposium on Interactive 3D Graphics*, pages 183–188, 1992.
16. C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti. Surround-screen projection-based virtual reality: the design and implementation of the

CAVE. In *Proc. ACM SIGGRAPH*, pages 135–142, 1993.
17. T. Delmarcelle and L. Hesselink. Visualizing second order tensor fields with hyperstreamlines. *IEEE Computer Graphics and Applications*, 13(4):25–33, 1993.
18. B. R. Donald and F. Henle. Using haptic vector fields for animation motion control. In *Proc. IEEE International Conference on Robotics and Automation*, pages 3435–3442, 2000.
19. N. I. Durlach and A. S. Mavor, editors. *Virtual Reality: Scientific and Technological Challenges*. National Academy Press, Washington, D.C., 1994.
20. A. L. Fuhrmann and M. E. Gröller. Real-time techniques for 3D flow visualization. In *Proc. IEEE Visualization*, pages 305–312, 1998.
21. B. Grant, A. Helser, and R. M. Taylor II. Adding force display to a stereoscopic head-tracked projection display. In *Proc. IEEE Virtual Reality Annual International Symposium*, pages 81–88, 1998.
22. J. Grosjean and S. Coquillart. Command & control cube: A shortcut paradigm for virtual environments. In *Proc. Eurographics Workshop on Virtual Environments*, Stuttgart, Germany, 2001.
Q7
23. C. Gunn and P. Marando. Experiments on the haptic rendering of constraints: guiding the user. In *Proc. Advanced Simulation Technology and Training Conference*, Melbourne, Australia, 1999.
24. B. Hannaford and J.-H. Ryu. Time domain passivity control of haptic interfaces. *IEEE Trans. Robotics and Automation*, 18(1):1–10, 2002.
25. M. Hutchins. A constraint equation algebra as a basis for haptic rendering. In *Proc. PHANToM Users Group Workshop*, Aspen, CO, 2000.
26. M. Ikits. Coregistration of pose measurement devices using nonlinear least squares parameter estimation. Technical Report UUCS-00-018, School of Computing, University of Utah, 2000.
27. M. Ikits, J. D. Brederson, C. D. Hansen, and J. M. Hollerbach. An improved calibration framework for electromagnetic tracking devices. In *Proc. IEEE Virtual Reality*, pages 63–70, Yokohama, Japan, 2001.
28. M. Ikits, J. D. Brederson, C. D. Hansen, and C. R. Johnson. A constraint-based technique for haptic volume exploration. In *Proc. IEEE Visualization*, pages 263–269, Seattle, WA, 2003.
29. M. Ikits, C. D. Hansen, and C. R. Johnson. A comprehensive calibration and registration procedure for the visual haptic workbench. In *Proc. Eurographics Workshop on Virtual Environments*, Zurich, Switzerland, 2003. submitted for publication.
30. Immersion Corporation. Medical simulators. http://www.immersion.com/
31. F. Infed, S. V. Brown, C. D. Lee, D. A. Lawrence, A. M. Dougherty, and L. Y. Pao. Combined visual/haptic rendering modes for scientific visualization. In *Proc. ASME Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 93–99, Nashville, TN, 1999.
32. H. Iwata and H. Noma. Volume haptization. In *Proc. IEEE Virtual Reality Annual International Symposium*, pages 16–23, Seattle, WA, 1993.
33. J. Kelso, L. E. Arsenault, and R. D. Kriz. DIVERSE: A framework for building extensible and reconfigurable device independent virtual environments. In *Proc. IEEE Virtual Reality*, pages 183–192, Orlando, FL, 2002.
34. Y. J. Kim, M. A. Otaduy, M. C. Lin, and D. Manocha. Six-Degree-of freedom haptic display using localized contact computations. In *Proc. IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 209–216, Orlando, FL, 2002.
35. G. L. Kindlmann. The teem toolkit, 2003. http://teem.sourceforge.net/
36. G. L. Kindlmann, D. M. Weinstein and D. A. Hart. Strategies for direct volume rendering of diffusion tensor fields. *IEEE Trans. Visualization and Computer Graphics*, 6(2):124–138, 2000.
37. M. Koutek and F. H. Post. The responsive workbench simulator: a tool for application development and analysis. In *Proc. International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, Plzen, Czech Republic, 2002.
38. A. Křenek. Haptic rendering of molecular flexibility. In *Proc. PHANToM Users Research Symposium*, Zurich, Switzerland, 2000.
39. D. A. Lawrence, C. D. Lee, L. Y. Pao, and R. Y. Novoselov. Shock and vortex visualization using a combined visual/haptic interface. In *Proc. IEEE Visualization*, pages 131–137, Salt Lake City, UT, 2000.
40. D. A. Lawrence, L. Y. Pao, A. M. Dougherty, M. A. Salada, and Y. Pavlou. Rate-hardness: a new performance metric for haptic interfaces. *IEEE Trans. Robotics and Automation*, 16(4): 357–371, 2000.
41. K. Lundin. Natural haptic feedback from volumetric density data. Master's thesis, Linköping University, Sweden, 2001.
42. A. Mascarenhas, S. Ehmann, A. Gregory, M. Lin, and D. Manocha. *Touch In Virtual*

Environments: Haptics and the Design of Interactive Systems, chapter 5: Six degree-of-freedom haptic visualization, pages 95–118. Prentice-Hall, 2002.
43. T. H. Massie. Design of a three degree of freedom force-reflecting haptic interface, bachelor's thesis, Massachusetts Institute of Technology, 1993.
44. T. H. Massie. Initial haptic explorations with the PHANToM: virtual touch through point interaction. Master's Thesis, Massachusetts Institute of Technology, 1996.
45. W. A. McNeely, K. D. Puterbaugh, and J. J. Troy. Six degree-of-freedom haptic rendering using voxel sampling. In *Proc. ACM SIGGRAPH*, pages 401–408, 1999.
46. M. Meyer and A. H. Barr. ALCOVE: design and implementation of an object-centric virtual environment. In *Proc. IEEE Virtual Reality*, pages 46–52, Houston, TX, 1999.
47. T. Miller and R. C. Zeleznik. An insidious haptic invasion: adding force feedback to the X desktop. In *Proc. ACM User Interface Software and Technology*, pages 59–64, 1998.
48. P. M. F. Nielsen, I. J. LeGrice, B. H. Smaill, and P. J. Hunter. Mathematical model of geometry and fibrous structure of the heart. In *American Journal of Physiology*, volume 260, pages H1365–H1378, 1991.
49. R. Y. Novoselov, D. A. Lawrence, and L. Y. Pao. Haptic rendering of data on unstructured tetrahedral grids. In *Proc. IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 193–200, 2002.
50. A. M. Okamura, J. T. Dennerlein, and R. D. Howe. Vibration feedback models for virtual environments. In *Proc. IEEE International Conference on Robotics and Automation*, pages 2485–2490, 1998.
51. M. A. Otaduy and M. C. Lin. Sensation preserving simplification for haptic rendering. *ACM Trans. Graphics*, 22(3):543–553, 2003.
52. L. Y. Pao and D. A. Lawrence. Synergistic visual/haptic computer interfaces. In *Proc. Japan/USA/Vietnam Workshop on Research and Education in Systems, Computation, and Control Engineering*, pages 155–162, 1998.
53. A. Petersik, B. Pflesser, U. Tiede, K. H. Höhne, and R. Leuwer. Haptic volume interaction with anatomic models at sub-voxel resolution. In *Proc. IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 66–72, 2002.
54. P. J. Rajlich. An object oriented approach to developing visualization tools portable across desktop and virtual environments. Master's Thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 1998.
55. ReachIn Technologies AB. Laparoscopic Trainer. http://www.reachin.se/.
56. M. Renz, C. Preusche, M. Pötke, H.-P. Kriegel, and G. Hirzinger. Stable haptic interaction with virtual environments using an adapted voxmap-pointShell algorithm. In *Proc. Eurohaptics*, 2001.
57. L. B. Rosenberg. Virtual fixtures: perceptual tools for telerobotic manipulation. In *Proc. IEEE Virtual Reality Annual International Symposium*, pages 76–82, 1993.
58. D. C. Ruspini, K. Kolarov, and O. Khatib. The haptic display of complex graphical environments. In *Proc. ACM SIGGRAPH*, pages 345–352, 1997.
59. S. E. Salcudean and T. D. Vlaar. On the emulation of stiff walls and static friction with a magnetically levitated input/output device. In *Proc. ASME Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 127–132, 1997.
60. J. K. Salisbury and C. Tarr. Haptic rendering of surfaces defined by implicit functions. In *Proc. ASME Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 61–67, 1997.
61. K. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles. Haptic rendering: programming touch interaction with virtual objects. In *Proc. ACM Symposium on Interactive 3D Graphics*, pages 123–130, Monterey, CA, Apr. 1995.
62. G. Sankaranarayanan, V. Devarajan, R. Eberhart, and D. B. Jones. Adaptive hybrid interpolation techniques for direct haptic rendering of isosurfaces. In *Proc. Medicine Meets Virtual Reality*, pages 448–454, Newport Beach, CA, 2002.
63. SensAble Technologies, Inc. FreeForm Modeling System. http://www.sensable.com/
64. L. Serra, T. Poston, N. Hern, C. B. Choon, and J. A. Waterworth. Interaction techniques for a virtual workspace. In *Proc. ACM Virtual Reality Software and Technology*, pages 79–90, Chiba, Japan, 1995.
65. S. Snibbe, S. Anderson, and B. Verplank. Springs and constraints for 3D drawing. In *Proc. PHANToM Users Group Workshop*, Dedham, MA, 1998.
66. S. S. Snibbe, K. E. MacLean, R. Shaw, J. Roderick, W. L. Verplank, and M. Scheeff. Haptic techniques for media control. In *Proc. ACM User Interface Software and Technology*, pages 199–208, Orlando, FL, 2001.

67. M. A. Srinivasan and C. Basdogan. Haptics in virtual environments: taxonomy, research status, and challenges. *Computer and Graphics*, 21(4):393–404, 1997.
68. D. R. Stevenson, K. A. Smith, J. P. McLaughlin, C. J. Gunn, J. P. Veldkamp, and M. J. Dixon. Haptic workbench: a multisensory virtual environment. In *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems*, pages 356–366, San Jose, CA, 1999.
69. P. S. Strauss and R. Carey. An object-oriented 3D graphics toolkit. In *Proc. ACM SIGGRAPH*, pages 341–349, Chicago, IL, 1992.
70. V. A. Summers, K. S. Booth, T. Calvert, E. Graham, and C. L. MacKenzie. Calibration for augmented reality experimental testbeds. In *Proc. ACM Symposium on Interactive 3D Graphics*, pages 155–162, Atlanta, GA, 1999.
71. T. Udeshi, R. Hudson, and M. E. Papka. Seamless multiresolution isosurfaces using wavelets. Technical report ANL/MCS-P801-0300, Argonne National Laboratory, 2000.
72. A. van Dam, A. S. Forsberg, D. H. Laidlaw, J. J. LaViola, Jr., and R. M. Simpson. Immersive VR for scientific visualization: a progress report. *IEEE Computer Graphics and Applications*, 20(6):26–52, 2000.
73. T. van Reimersdahl, F. Bley, T. Kuhlen, and C. Bischof. Haptic rendering techniques for the interactive exploration of CFD datasets in virtual environments. In *Proc. Eurographics Workshop on Virtual Environments*, pages 241–246, Zurich, Switzerland, 2003.
74. D. M. Weinstein, G. L. Kindlmann, and E. Lundberg. Tensorlines: advection-diffusion based propagation through diffusion tensor fields. In *Proc. IEEE Visualization*, pages 249–253, San Francisco, CA, 1999.
75. M. Woo, J. Neider, T. Davis, and D. Shreiner. *OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2*. Addison Wesley, 1999.
76. S. Zhang, C. Demiralp, D. Keefe, M. DaSilva, D. H. Laidlaw, B. D. Greenberg, P. Basser, C. Pierpaoli, E. Chiocca, and T. Deisboeck. An immersive virtual environment for DT-MRI volume visualization applications: a case study. In *Proc. IEEE Visualization*, pages 437–440, San Diego, CA, 2001.
77. C. B. Zilles and J. K. Salisbury. A constraint-based god-object method for haptic display. In *Proc. IEEE International Conference on Intelligent Robots and Systems*, pages 146–151, Pittsburgh, PA, 1995.

AUTHOR QUERIES

Q1 Au: what is the phanton?
Q2 Au: as useful as what?
Q3 Au: please speel out NURBS?
Q4 Au: servo ok?
Q5 Au: ok?
Q6 Au: comma ok?
Q7 Au: pagge numbers available?