The Proxy Chain Method and Its Application to Scientific Visualization

Milan Ikits Charles D. Hansen

Scientific Computing and Imaging Institute, University of Utah

Abstract

We present a method for combining multiple point-based constraints in haptic programming environments. Instead of using a single proxy point for haptic feedback, the method maintains a separate proxy for each constraint. The reaction force is computed by linking the proxies in a chain. Constraints are applied in sequential order, such that the proxy found in the current step becomes the probe for the next step in the chain. The advantage of the method over previous approaches is that the constraints are maintained precisely and the output is well-defined. We illustrate the method with examples from the domain of 3D scientific data visualization. Finally, we present the results of an experiment conducted to quantify the contribution of haptic guidance in two representative vector field exploration tasks.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality; I.3.6 [Computer Graphics]: Methodology and Techniques—User Interfaces; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Haptic I/O

Keywords: haptic rendering, scientific data visualization, human-computer interaction

1 Introduction

The extension of constraint-based surface rendering algorithms [26] to volumetric data sets has enabled the development of intuitive haptic data rendering modes for interactive visualization applications [7, 14]. Volumetric constraints have been shown useful as a general mechanism for creating strong directional and shape cues, in addition to secondary haptic effects, such as friction and texture, to aid the exploration of scalar, vector, and tensor data.

Previous approaches to using constraints for developing haptic data rendering modes were based on incrementally moving a single proxy point in the data [7, 13, 14, 25]. It has been demonstrated that by combining multiple constraints, a variety of haptic modes can be implemented, including modes to display volumetric viscosity, surface stiffness and friction, as well as vector and tensor field orientation. However, the majority of the approaches based on a single proxy point are limited to constraints that are expressed along orthogonal reference directions [7, 14, 25]. A recent method overcomes the orthogonal reference frame limitation, but fails to maintain the constraints precisely during the proxy update step [13].

We present the proxy chain as a general solution to combining multiple volumetric constraints in haptic programming environments. The advantage of the method over previous approaches [6, 13] is that the constraints are maintained precisely and the output is well-defined. We illustrate the algorithm with examples from the domain of 3D scientific data visualization. Finally, we also present the results of an experiment conducted to quantify the contribution of haptic guidance in two representative vector field exploration tasks.

2 Interactive Data Exploration

Haptic feedback has three related uses in scientific visualization applications. First, a rich lexicon of kinesthetic and tactile cues can be used to display the salient features in the data. For example, surfaces can be rendered with additional information overlaid to indicate discontinuities, transitions, and inhomogeneous regions [14, 23]. Haptically guided data exploration can aid the user in various tasks, such as finding and measuring features or investigating the relationship between several quantities in multi-field data sets [7, 13]. Second, force feedback can improve user performance in data manipulation tasks, such as molecular docking [3, 15] and interactive segmentation [25]. Third, the associated user interface can benefit from haptic assistance. For example, guiding forces can be added to aid the user during selection and manipulation of interface elements or to indicate feasible configurations of a selected object [11, 19].

The traditional approach to using haptic feedback for data exploration is based on mapping data variables directly to force and torque components [1, 8, 10]. Direct mapping works well for simple tasks, such as finding locally extremal regions in scalar data sets. Force and torque fields have also be used to provide guidance towards specific features in the data [2, 5, 17]. However, force fields have limited expressive power and also suffer from occlusion and instability problems [7, 26].

Recent research has extended the constraint-based surface rendering technique to volumetric data sets [7, 14]. The constraint-based approach has the advantage that it provides a unified basis for rendering a variety of data modalities. Furthermore, various haptic effects can be created by dynamically modifying the constraints based on local properties of the data.

Constraint-based data rendering methods are composed of four components that are executed during every frame of the haptic servo loop [7]. First, data elements corresponding to the position of the probe and the proxy are located in the data mesh. Second, local data measures are computed by interpolating data elements around the current proxy location. Third, the proxy is displaced in the volume, depending on the position of the probe relative to the proxy, the computed local measures of the data, as well as haptic transfer functions and rendering parameters. Finally, the reaction force is obtained using a spring coupler between the proxy and the probe. Note that with the exception of the force computation component, the components may be executed several times in a frame as part of an iterative procedure, as discussed in Section 4.

There are three general requirements that any haptic data rendering technique must satisfy. First, to provide the user with a consistent experience, the displayed haptic cues cannot result in conflicting visual feedback. For example, when exploring streamlines in vector field data, the path traversed by the haptic probe has to match the visual representation of the selected streamline. Second, the technique should provide the user with a way to freely explore the available workspace. This can be achieved by changing the feedback mode based on user input, indicated either through a separate input device or the force from the haptic interface. Third, in addition to creating strong haptic cues for guidance, the system has to provide visual or haptic indication whenever the user leaves a selected feature of interest in the data.

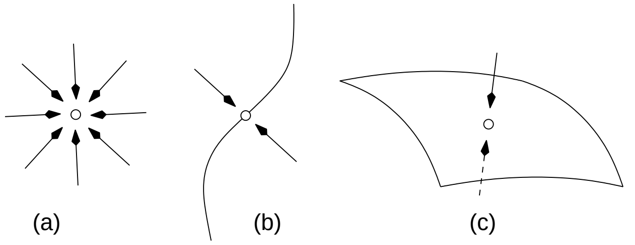


Figure 1: Basic types of 3D constraints: (a) point, (b) curve, and (c) surface constraint.

3 Haptic Constraints

Point-based constraints are essential components of haptic programming packages. A variety of haptic rendering algorithms have been developed for the unambiguous and stable display of geometric models based on constraining the motion of the probe to the surface of the model [22,24,26]. Algorithms for displaying friction typically use a secondary constraint to account for the stick-slip motion of the fingers on the surface of an object [18].

General support for constraints was first incorporated into the MAGMA scene graph API in the form of a haptic constraints library [6]. The library included a class hierarchy for point, line, and surface constraints and also provided a mechanism for combining multiple constraints. In the constraint pipe algorithm, constraints are applied repeatedly until all of them are satisfied or the maximum number of iterations is exceeded. Since in certain situations it is impossible to satisfy multiple constraints with a single proxy point, the output of the algorithm is not always well-defined.

A similar iterative procedure was proposed for finding the set of active planar constraints in the god-object method [26]. The iteration terminates when no new constraint can be added to the system. The location of the proxy is updated by minimizing the distance between the probe and the proxy subject to the set of active constraints using Lagrange multipliers. However, it is not clear how to extend the constrained optimization method to nonlinear, implicit, and overlapping constraints.

The OpenHaptics Toolkit uses the OpenGL feedback buffer to capture geometric point, line, and polygon primitives for haptic rendering [9]. The captured polygon primitives can be used in unilateral or bilateral constraint mode. When multiple overlapping constraints are specified, the API constrains the proxy according to decreasing order of dimensionality. Thus, the system provides a method for restricting probe motion to a lower dimensional constraint embedded into a higher dimensional constraint. For example, the probe can be attracted to a point and constrained to a plane simultaneously.

Recently, a force-based approach was developed for combining multiple constraints represented by linear haptic primitives [13]. At every step, the position of the proxy is updated by balancing the force from the probe with the forces from the primitives. The primitives are placed automatically according to the local properties of the data and the selected haptic mode. Since the proxy obtained from the force balance equation may not satisfy any of the constraints, it is not possible to implement precise directional haptic modes using the force-based approach.

Volumetric constraints can be augmented with snap and drag parameters [7]. Snapping is used for activating and deactivating a constraint based on its proximity to the probe. Dragging facilitates precise repositioning of the constraint by damping unwanted oscillations from the probe. The combination of snapping and dragging can also be used for implementing simple friction and texture effects.

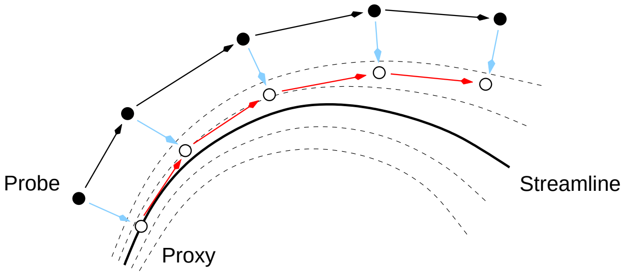


Figure 2: Exploring a streamline in a vector field using the follow mode and haptic line primitives. Due to the poor numerical accuracy of the linear approximation, the proxy can drift from the selected streamline, resulting in conflicting visual cues. The error is most noticeable in regions of high field divergence.

4 Nonlinear Constraints

In three dimensional space, the proxy can be locally constrained in three independent directions. As illustrated in Figure 1, depending on the number of constrained degrees of freedom, a point, curve, or surface constraint is obtained.

Typically, constraints are expressed in a linear form, either in a local reference frame [7] or using haptic primitives [13]. The linear form is appropriate only when the corresponding visual feedback is also built on the same representation. To illustrate the problem, consider exploration of streamlines in vector field data using the follow mode [4] and haptic line primitives [13]. As illustrated in Figure 2, due to the poor numerical accuracy of the linear approximation, the proxy can drift from the selected streamline during exploration, resulting in conflicting visual cues. The error depends on the divergence and vorticity of the vector field and the speed of exploration. The line primitive is equivalent to using an Euler integrator for solving the ODE that describes the particle advection process,

$$\vec{p}_k = \vec{p}_{k-1} + \vec{v}(\vec{p}_{k-1})\,\Delta t \tag{1}$$

where $\vec{p}_k$ is the location of the proxy at time step k, $\vec{v}(\vec{p})$ is the field vector at position $\vec{p}$, and Δt is a function of $\vec{x}_k$, the position of the probe at time step k, relative to $\vec{p}_{k-1}$, the position of the proxy at time step $k-1$.

$$\Delta t = \frac{\vec{v}(\vec{p}_{k-1}) \cdot [\vec{x}_k - \vec{p}_{k-1}]}{\|\vec{v}(\vec{p}_{k-1})\|^2} \tag{2}$$

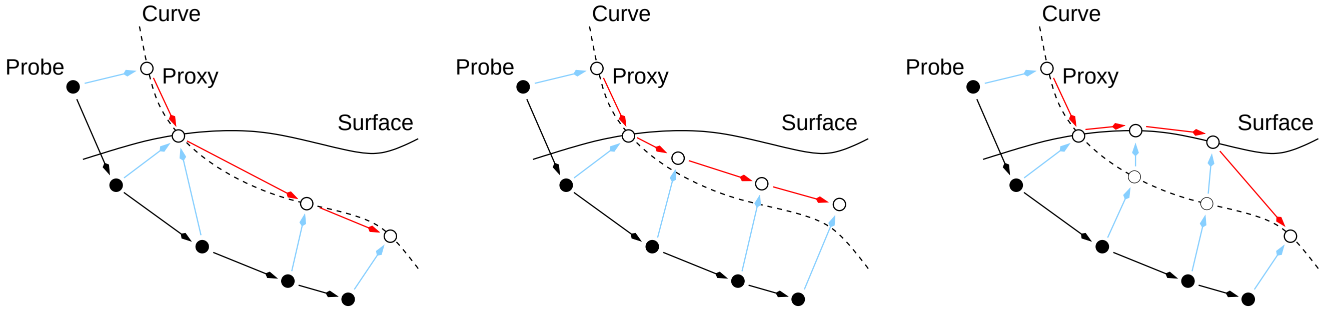
In practice, Euler integration is avoided for graphical display because of its poor numerical accuracy. Higher order schemes with adaptive stepsize control are preferred instead [21]. Hence, assuming that a function $\vec{f}_V(\vec{p},\Delta t)$ is provided, which for vector field V solves the particle advection problem from starting position $\vec{p}$ and step size Δt, Equation 1 can be substituted with an iterative procedure,

$$\vec{p}_{k,j+1} = \vec{p}_{k,j} + \vec{f}_V(\vec{p}_{k,j}, \frac{\vec{v}(\vec{p}_{k,j}) \cdot [\vec{x}_k - \vec{p}_{k,j}]}{\|\vec{v}(\vec{p}_{k,j})\|^2}) \tag{3}$$

where $\vec{p}_{k,0} = \vec{p}_{k-1}$. The iteration terminates when the residual vanishes, which corresponds to a local extremum of the probe-streamline distance function. Note that the advection function is defined reversible.

$$\vec{f}_V(\vec{p}, -\Delta t) = \vec{f}_{-V}(\vec{p}, \Delta t) \tag{4}$$

If the algorithm fails to converge, which may happen for example at critical points in the field, it falls back either to the linear approximation or, if detected, the location of the critical point. The experiment described in Section 7 uses a third-order Runge-Kutta



(a)

(b)

(c)

Figure 3: Combining a penetrable isosurface constraint and a streamline curve constraint. In this example, the probe is constrained to the curve until penetrating the surface. (a) When using the constraint pipe method, the proxy stays fixed at the intersection of the two constraints until the distance between the probe and the proxy reaches a user-defined threshold and the surface constraint is deactivated. Note that depending on the relative orientation of the two constraints, the surface constraint may be deactivated even if the probe only slightly penetrates the surface. (b) With the force-based approach, the proxy is updated by balancing the forces from the probe, the surface constraint and the curve constraint. Note that after penetrating the surface, the proxy will drift from both constraints. (c) By keeping a separate proxy for each constraint, the probe can be constrained to the curve and the surface simultaneously. The surface constraint is deactivated when the distance between the curve proxy and the surface proxy reaches a user defined limit, resulting in a better approximation to a penetrable surface of finite depth.

method with fourth-order adaptive stepsize control as the advection function $\vec{f}_V$.

A similar iterative method is used for constraining the motion of the proxy to isosurfaces [7, 22]. To ensure that the proxy stays on the surface, the linear estimate is refined using Newton-Raphson iteration along the gradient direction,

$$\vec{p}_{k,j+1} = \vec{p}_{k,j} - \frac{\nabla s\,(\vec{p}_{k,j})\,[s(\vec{p}_{k,j}) - s_0]}{\|\,\nabla s\,(\vec{p}_{k,j})\,\|^2} \tag{5}$$

where $s(\vec{p})$ and $\nabla s(\vec{p})$ are the field value and gradient at position $\vec{p}$, respectively. The linear estimate is obtained by projecting the probe to the tangent plane at the proxy.

$$\vec{p}_{k,0} = \vec{x}_k - \nabla s\,(\vec{p}_{k-1})\frac{\nabla s\,(\vec{p}_{k-1})\cdot[\vec{x}_k - \vec{p}_{k-1}]}{\|\nabla s\,(\vec{p}_{k-1})\|^2} \tag{6}$$

The refinement is terminated when the residual $[s(\vec{p}) - s_0]$ is sufficiently small. The algorithm falls back to the linear approximation when the refinement does not converge before reaching the maximum number of iterations permitted.

5 The Proxy Chain Method

To illustrate the problem of combining multiple intersecting or overlapping constraints, consider the combination of a penetrable isosurface constraint and a streamline curve constraint. In the example shown in Figure 3, the data probe is constrained to the curve until the proxy penetrates the surface. The goal is to follow to curve inside the surface and simultaneously provide haptic cues about the penetration depth. When using the constraint pipe method [6], the proxy stays fixed at the intersection of the two constraints until the distance between the probe and the proxy reaches a user-defined threshold and the surface constraint is deactivated. As shown in Figure 3(a), both constraints remain satisfied, but no haptic cues are generated indicating the penetration depth into the surface. In addition, depending on the relative orientation of the two constraints, the surface constraint may be deactivated even if the probe only slightly penetrates the surface. With the force-based approach [13], shown in Figure 3(b), the location of the proxy is updated by balancing the forces from the probe, the surface constraint and the curve constraint. Since the update step uses only local information from the two fields and it attempts to globally balance the penalty forces obtained from violating the curve and the surface constraint, the proxy will drift from both constraints. In fact, there is no guarantee that the proxy will satisfy any of the constraints. Thus, it is not possible to continue exploring the original streamline curve after the surface constraint is deactivated, because the proxy will be located on a different streamline.

There is no general proxy update strategy that works for all applications [6]. One possible choice is to always select the constraint closest to the probe [9]. This approach is useful for building simple compound constraints that feel continuous, such as a curve constraint from connected line segments or a surface constraint from triangle primitives. However, the approach suffers from the same problems as the force field method [26]. In addition, it is not possible to implement intersecting constraints without adding lower dimensional primitives to the scene.

The solution to the problem of combining a penetrable isosurface constraint and a streamline curve constraint is to maintain two separate proxies for the constraints. The proxies are updated in series during the proxy update step to obtain the desired haptic effect. Thus, instead of finding a single proxy location that attempts to satisfy all of the constraints simultaneously, the proxies form a chain, such that the proxy found in the current step becomes the probe for the next step in the chain. As shown in Figure 3(c) by keeping a separate proxy for each constraint, the probe can be constrained to the curve and also provide feedback about the penetration depth. Thus, the proxy update step is composed of two parts. First, the curve proxy is updated using the new probe position. Next, the surface proxy is updated using the new curve proxy as the probe. The surface constraint is deactivated when the distance between the curve proxy and the surface proxy reaches a user defined limit, resulting in a better approximation to a penetrable surface of finite depth.

6 Examples

6.1 Anisotropic Friction Mode for Mixed Scalar/Vector Data

We are interested in developing combined visual and haptic rendering modes for coregistered scalar and vector data sets. See Figure 4 for examples. Using the proxy chain method, data probing tools can be easily augmented with multiple haptic constraints. Since there is no restriction on the relative orientation of the reference frames of

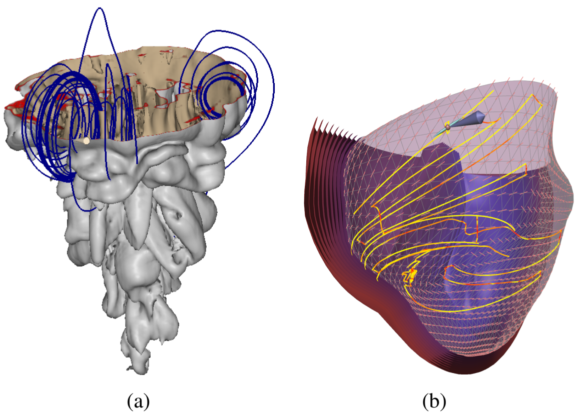


Figure 4: Examples of combined visual and haptic exploration of coregistered scalar and vector data. (a) Exploring the interaction between the density field and the velocity field in a CFD data set. (b) Haptic rendering of epicardial muscle fibers [7].

the proxies, arbitrary combinations of nonlinear constraints are supported. However, not all combinations result in useful and intuitive haptic feedback modes.

We developed an anisotropic friction mode that provides haptic cues about the interaction between a 3D scalar field and a 3D vector field. As shown in Figure 5(a), the anisotropic friction mode uses three proxies in the chain. Each proxy is updated according to a set of motion rules and transfer functions, as described in [7]. The first proxy $\vec{p}_s$ restricts probe motion to a selected isosurface of the scalar field. The second proxy $\vec{p}_c$ provides directional cues by adding a friction effect perpendicular to the tangential component of the vector field. The third proxy $\vec{p}_f$ implements an additional friction term that is modulated according to the orientation of the field relative to the surface. The simulated effect is that the more perpendicular the field vector $\vec{v}$ to the surface, the more difficult it is to drag the probe along the streamline. Hence, the drag threshold for the third proxy τ_f is a function of the field vector $\vec{v}$ and the surface gradient ∇s at $\vec{p}_f$,

$$\begin{aligned} \tau_f &= u\,\tau_{min} + (1-u)\,\tau_{max} && (7)\\ u &= \cos\theta && (8)\\ &= [\,\nabla s \cdot \vec{v}\,]\;\sigma(\|\nabla s\|\|\vec{v}\|) && (9) \end{aligned}$$

where τ_{min} and τ_{max} are parameters for the minimum and maximum frictional drag threshold, respectively, and $\sigma(x)$ is a safe normalization term that ensures that the interpolation factor u reduces to zero in ill-defined regions of the fields.

6.2 Constrained Haptic Modes for 3D User Interfaces

User interface elements can also benefit from constrained haptic feedback. For example, a 3D widget can provide intuitive and precise control over the position and orientation of cutting planes in a volume. We implemented a haptic widget that includes a spherical constraint and three planar constraints, as shown in Figure 6(a). The active constraints are ordered in the proxy chain according to the current mode of operation. For example, initially the user can snap the probe to the cutting plane to explore the data values on its surface. Pressing a button activates translation mode. In this mode, the cutting plane can be translated along three orthogonal directions and their combinations. A plane is activated based on the distance of the probe from the corresponding proxy location. Note that the order of the plane constraints in the chain is not important, because the planes are orthogonal to each other.

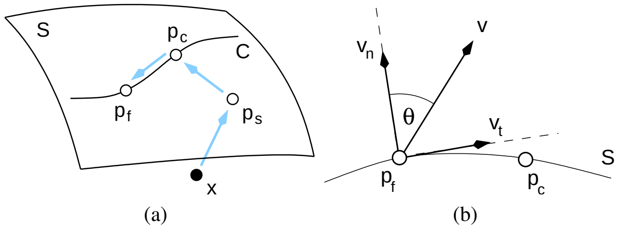


Figure 5: (a) Anisotropic friction mode for exploring mixed scalar and vector fields. By linking three proxies in a chain, the probe can be constrained to an isosurface of the scalar field and a streamline defined by the tangential component of the vector field. (b) The third proxy implements a friction effect that is modulated according to the orientation of the vector field relative to the surface. The field vector is decomposed into tangential and normal components. The more perpendicular the field vector is to the surface, the more difficult it is to follow the streamline with the probe. Note that all proxies are located on the surface.

When the probe is in proximity of the sphere, rotational mode is activated. In rotational mode, the sphere constraint is applied last in the proxy chain after the planar constraints. The advantage of using the proxy chain is that both axis-constrained and unconstrained rotational modes can be implemented without having to add embedded lower dimensional circle, line, and point constraints to the widget. In addition, it is easy to switch between modes of operation by activating, deactivating, and reordering constraints in the chain.

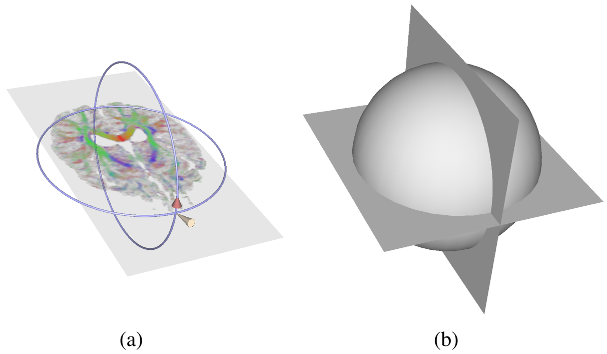


Figure 6: (a) A haptic widget for specifying the position and orientation of cutting planes in a volume. (b) The corresponding active constraints in the widget. In the configuration shown above, only two out of the three planes are active. The widget provides support for both axis-constrained and unconstrained translations and rotations.

7 Contribution of Haptic Guidance to Vector Field Exploration Tasks

To quantify the contribution of haptic feedback in data exploration tasks, a controlled experiment was conducted. The purpose of the study was to evaluate user performance for two representative tasks with and without haptic guidance and to examine the difference in difficulty between exploring 2D and 3D data sets. The chosen task domain was vector field visualization. The tasks were designed based on related experiments from the literature [12, 20]. To simplify the tasks, only a single haptic constraint was included, except for one task, where the proxy chain method was used for combining a streamline constraint with a plane constraint, as illustrated in Figure 3(c).

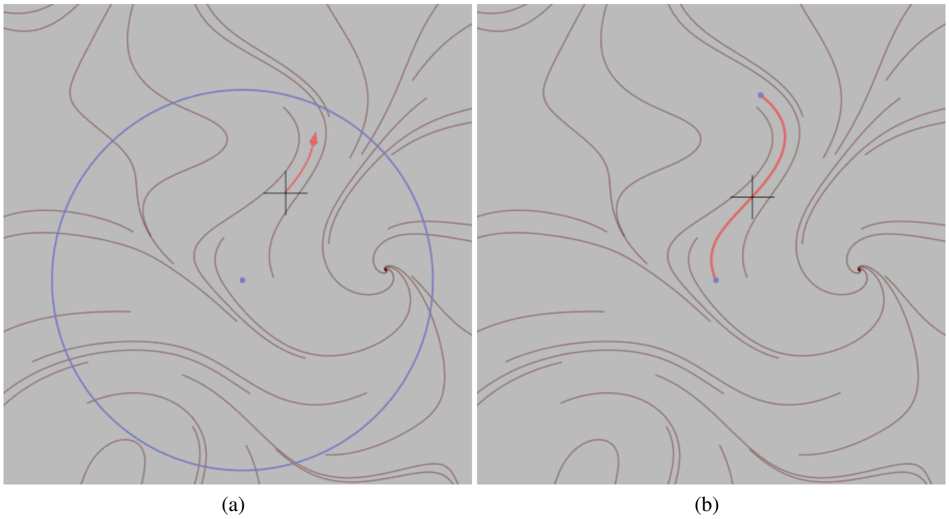


Figure 7: Illustration of the 2D vector field exploration tasks. (a) During the advection task, subjects followed a local streamline indicator with the cursor from the starting position until the cursor hit the edge of the circle placed around the initial position. (b) During the tracing task, subjects followed a static streamline segment with the cursor from one end to the other.

7.1 Experimental Procedure

For each task, subjects were presented with images of synthetic vector fields. Each image consisted of a sparse set of streamlines that indicated the global behavior of the field (Figures 7 and 8). In addition to the global view, subjects were required to use the haptic probe to learn more about the local properties of the data.

A stereoscopic desktop workstation and a SensAble PHANToM 3.0L was used for the experiment. Images for the 3D tasks were displayed in stereo using CrystalEyes LCD shutter glasses. The haptic device was positioned on the right hand side of the monitor. The stylus was represented with a crosshair cursor within the visual workspace. A keyboard was also added to collect input from the users without disturbing the motion of the haptic probe.

For the 2D advection task, subjects were asked to follow a local indicator, as shown in Figure 7(a). The starting position was selected so that the correct path would not match one of the global streamlines. The task was completed when the cursor hit the edge of the circle placed around the starting position. The radius of the circle was chosen such that the correct path had a constant length for every trial. A short streamline segment was advected from the cursor position with an arrowhead placed at the end, indicating the local direction of the flow. Users were instructed to follow the local indicator as accurately as possible in a single sweeping motion. In addition, they were instructed to complete the task as quickly as possible and were given a 10 second time limit.

For the 2D tracing task, subjects had to follow a static path with the cursor, as illustrated in Figure 7(b). The path was represented by a red streamline segment originating from the starting point. The task was completed when the cursor reached the target at end of the segment. Users were instructed to trace the path with the cursor as accurately as possible and to keep the cursor on the streamline all the time. Completion time was restricted to 10 seconds for this task as well.

The 3D tasks had a similar structure to the 2D tasks, as shown in Figure 8. The most important difference was that a plane target was used for the advection task rather than a hemisphere. A plane was chosen, because intersecting a semitransparent hemisphere was deemed too difficult during the pilot studies. The plane was always oriented as a ground plane and was textured with a checkerboard pattern. Several other measures were taken to improve depth perception of the scene: depth cuing using fog and lighting cues was added, and the starting position was chosen to minimize occlusion of the path by the global streamlines. In addition, the path and the viewing orientation were selected such that most of the cursor motion was parallel with the image plane of the monitor. Finally, a haptic cue was added to indicate the intersection of the cursor with the plane.

Two different haptic stimuli were used in the experiment. For the ISO condition, a slight isotropic drag force was added using a point constraint. The isotropic feedback was helpful, because it kept the probe in the same location when users released the stylus and stabilized the motion of the probe during the tasks. For the ANISO condition, probe motion was restricted to the correct streamline using a nonlinear constraint, as described in Section 4. The anisotropic feedback helped users to keep the probe on the path. However, they still had to make sure the cursor visually intersected the streamline, because with some effort it was still possible to move the cursor off the path due to the limited stiffness of the haptic interface.

Fifteen male and five female participants were recruited from the University of Utah School of Computing student and staff pool. The experiment did not require the subjects to know anything about flow visualization techniques. The age of the participants varied between 22 and 44 years and all of them were right handed.

Task completion time and position error were used as performance measures for the evaluation. For the advection tasks, position error was defined as the magnitude of the difference between the correct intersection point and the one found by the user. For the tracing tasks, position error was defined as the average deviation from the correct path. The deviation was computed as the distance between the cursor and the closest point on the curve.

Not all trials were considered in the analysis. Trials with unusually high completion times or position errors were discarded. These trials correspond to the cases when users got lost in the data. In total, 70 of the 1600 trials were rejected based on cut-off error limits (30 mm for 2D and 100 mm for 3D) and time limits (10 and 15 seconds for the advection tasks and 12 and 17 seconds for the tracing tasks). The time cutoffs for the tracing tasks were increased, because users spent more time completing these tasks. The means and standard deviations of the performance measures for each user were input for statistical analysis of the data.

In addition to the measurements, informal feedback from each subject was collected after the experiment in the form of a questionnaire. The questions included whether subjects found the 3D

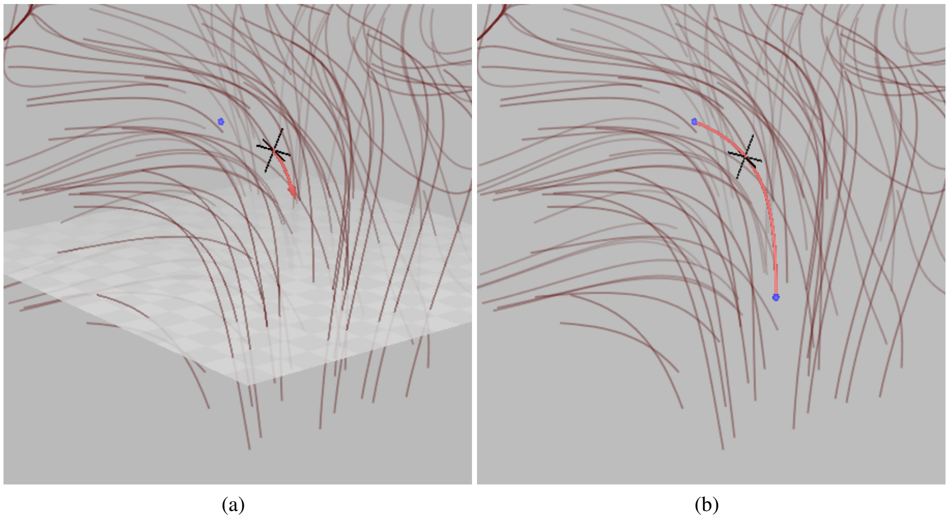


Figure 8: Illustration of the 3D vector field exploration tasks. (a) During the advection task, subjects followed a local streamline indicator with the cursor from the starting location until the cursor hit the ground plane placed below the starting position in the data. (b) During the tracing task, subjects followed a static streamline segment with the cursor from one end to the other.

trials more difficult than the 2D trials and if they used any particular strategy for completing the tasks.

7.2 Results and Analysis

Three different methods were used for analyzing the data. Pairwise t-tests for the means and standard deviations of both measures between the ANISO and ISO conditions provided direct numerical comparison of subject performance. The t-tests were augmented with graphical comparison of the condition means along with 95% confidence intervals. Based on the graphical comparison, two means can be considered statistically different, if the confidence intervals do not overlap. The graphs for the 2D and 3D tasks are shown in Figures 9 and 10, respectively. Finally, to examine the correlation between position error and completion time, per subject means were plotted for each task, as shown in Figure 11.

One would think that the lack of continuous global feedback made the advection task more difficult than the tracing task. However, subjects spent more time on the tracing task, due to the direct visual feedback provided about their performance. Correspondingly, the subjects felt that the tracing tasks were more difficult, which was also indicated by the post-experiment questionnaire. Since for the advection tasks performance feedback was limited, making a conscious effort to find the optimal path was not always possible. The experimental measurements reflect the observation of the users, because the tracing tasks took more time to complete with significantly better accuracy.

The results of the pairwise analysis indicate that for all four tasks the ANISO condition improves both positioning accuracy and completion time significantly. The pooled analysis in Figures 9 and 10, however, shows no significant difference between task completion times, except for the 3D tracing task. We believe that the results express the fact that the connection between the graphical and haptic feedback is easier to establish for the tracing tasks. In addition, many users reported that the difference between the ANISO and ISO conditions is more noticeable for the 3D tasks than the 2D tasks. Thus, the two conditions were easiest to distinguish for the 3D tracing task. However, since there was no strict control established for the timing of the tasks and the advection task had limited feedback on task performance, general conclusions should not be drawn from the timing data, other than comparing the difficulty of the 2D and 3D tasks.

All users indicated in the post-experiment survey that the 3D tasks were significantly more difficult than the 2D versions. Pairwise t-tests between the accuracy of the 2D and 3D tracing tasks confirm this claim, even with haptic feedback (t(19)=7.38, $p < 0.001$). Most participants complained about the effort needed to find the intersection of the 3D cursor with the streamline, due to the limited depth information provided by the graphical feedback. For intersection tasks, better local feedback could be provided by orthogonal transparent planes. However, finding the right balance between occlusion and transparency is not straightforward and is also subjective.

For a few subjects depth perception was problematic. One subject had difficulty seeing the difference in depth between the starting and target locations on the streamline. Two other users experienced eye strain from focused viewing of the stereo imagery.

An important issue was speed of execution. Many users felt that with smoother and faster motions they performed better, even without haptic feedback. One subject, however, noticed a conditioning effect when several trials with the ANISO condition were present in a row.

In general, users appreciated the haptic guidance for both tasks. Several subjects reported that they used the strategy of slowly determining at the beginning of the trial if the guiding forces were present or not and using them to quickly complete the task. In contrast, others found that they relied completely on visual feedback, but felt that some trials were easier than others. Most users followed the local indicator to accomplish the advection tasks, however, they considered it only a rough guide. Only one user reported on the usefulness of global feedback for the 2D advection task. However, for the 3D case, the global feedback was found inefficient by the majority of the subjects.

Another important question is whether there was correlation between the timing and accuracy measures. The most significant correlation was established for the 2D tracing task with the ISO condition (Pearson: $-0.704, p < 0.001$). The general trend in the distributions corresponding to the ISO condition is that whenever users tried to accomplish the task faster, accuracy was degraded. The trend was less significant for the ANISO condition.

The outcome of the experiment confirms that the human visual and fine motor skills excel in two dimensions, especially when proper visual feedback is available. The importance of constrained feedback in three dimensional precision tasks is reflected by the

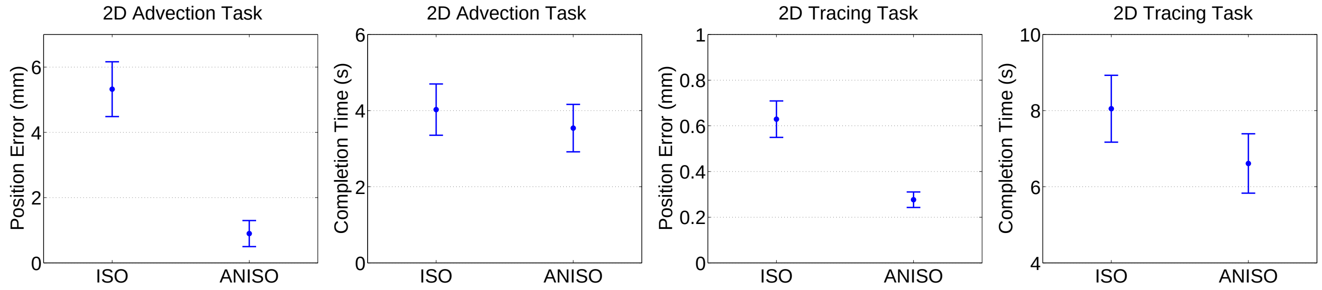


Figure 9: Mean position errors and task completion times for the 2D tasks. The graphs show that both measures are lower for the ANISO condition than for the ISO condition. However, there is no significant difference between task completion times.c

large differences between the ANISO and ISO conditions with respect to both positioning accuracy and timing of the tasks. When visual feedback on user performance was available, subjects made efforts to complete the tasks more accurately. Best results, however, were obtained when both continuous visual and haptic feedback were provided to the users.

8 Summary and Future Work

We have presented a method for combining multiple constraints in haptic programming environments. We illustrated the method with examples from the domain of 3D scientific visualization. A particular advantage of the approach is that intersecting and overlapping constraints are treated in a unified framework. Furthermore, the algorithm can handle the combination of nonlinear constraints in contrast to a previous method [13].

A major limitation of existing constraint-based data rendering techniques is that the coupling between the visual and haptic feedback is restricted to a single point in the data. In contrast, force field methods have been extended to 6DOF haptic devices and various probe shapes [16]. However, it is not clear whether constraining torques would provide any benefit in data exploration tasks.

The presented experiment is an initial step towards quantifying the contribution of haptic guidance in data exploration tasks. In the future, we plan to study the utility of haptic feedback in complex data exploration and manipulation tasks, and develop additional haptic user interfaces elements for immersive visualization applications. We are also interested in extending the constrained-based approach to 6DOF haptic devices, as well as multiresolution and point data sets.

Acknowledgments

The authors thank Sarah H. Creem-Regehr, Department of Psychology, University of Utah, for helpful discussions on the design of the experiment and providing access to statistical data analysis tools. The authors are also grateful to the volunteers who participated in the experiment. The Teem toolkit (`http://teem.sf.net/`) was used for data probing and manipulation. The CFD data set is courtesy of J. Bell and V. Beckner, Center for Computational Sciences and Engineering, Lawrence Berkeley National Laboratory. The heart data set is courtesy of the Bioengineering Institute at the University of Auckland. The diffusion tensor data set is courtesy of Gordon Kindlmann and Andrew Alexander, W. M. Keck Laboratory for Functional Brain Imaging and Behavior, University of Wisconsin-Madison. Support for this research was provided by NSF grant ACI-9978063 and the DOE Advanced Visualization Technology Center.

References

[1] R.S. Avila and L.M. Sobierajski. A haptic interaction method for volume visualization. In *Proc. IEEE Visualization*, pages 197–204, San Francisco, CA, Oct. 1996.

[2] D. Bartz and Ö. Gürvit. Haptic navigation in volumetric datasets. In *Proc. PHANToM Users Research Symposium*, Zurich, Switzerland, July 2000.

[3] 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, Aug. 1990.

[4] 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, San Francisco, CA, Apr. 2000.

[5] J.P. Fritz. Haptic rendering techniques for scientific visualization. Master's thesis, Department of Electrical Engineering, University of Delaware, Newark, DE, Dec. 1996.

[6] M. Hutchins. Software components for haptic constraints. In *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems*, pages 423–432, San Jose, CA, Jan. 2000.

[7] 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, Oct. 2003.

[8] 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, Nov. 1999.

[9] B. Itkowitz, J. Handley, and W. Zhu. The OpenHaptics™ toolkit: A library for adding 3D Touch™ navigation and haptics to graphics applications. In *Proc. World Haptics Conference*, Pisa, Italy, Mar. 2005.

[10] H. Iwata and H. Noma. Volume haptization. In *Proc. IEEE Virtual Reality Annual International Symposium*, pages 16–23, Seattle, WA, Sept. 1993.

[11] R. Komerska and C. Ware. Haptic task constraints for 3D interaction. In *Proc. IEEE Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 270–277, Los Angeles, CA, Mar. 2003.

[12] D.H. Laidlaw, R.M. Kirby, C.D. Jackson, J.S. Davidson, T.S. Miller, M. da Silva, W.H. Warren, and M.J. Tarr. Comparing 2D vector field visualization methods: A user study. *IEEE Transactions Visualization and Computer Graphics*, 11(1):59–70, 2005.

[13] K. Lundin, B. Gudmundsson, and A. Ynnerman. General proxy-based haptics for volume visualization. In *Proc. World Haptics Conference*, Pisa, Italy, Mar. 2005.

[14] K. Lundin, A. Ynnerman, and B. Gudmundsson. Proxy-based haptic feedback from volumetric density data. In *Proc. Eurohaptics*, Edinburgh, United Kingdom, July 2002.

[15] R. Maciejewski, S. Choi, D.S. Ebert, and H.Z. Tan. Multi-model perceptualization of volumetric data and its application to molecular docking. In *Proc. World Haptics Conference*, Pisa, Italy, Mar. 2005.

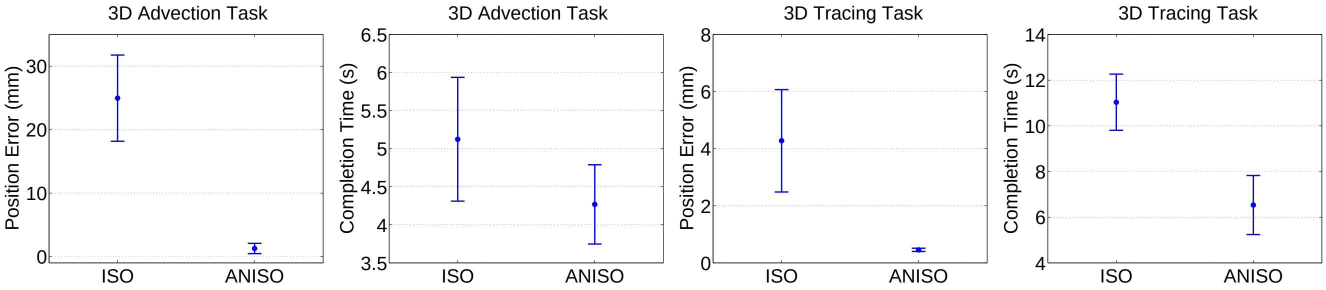


Figure 10: Mean position errors and task completion times for the 3D tasks. The graphs show that both measures are lower for the ANISO condition than for the ISO condition. Completion time is significantly lower for the ANISO condition for the tracing task, but not for the advection task.

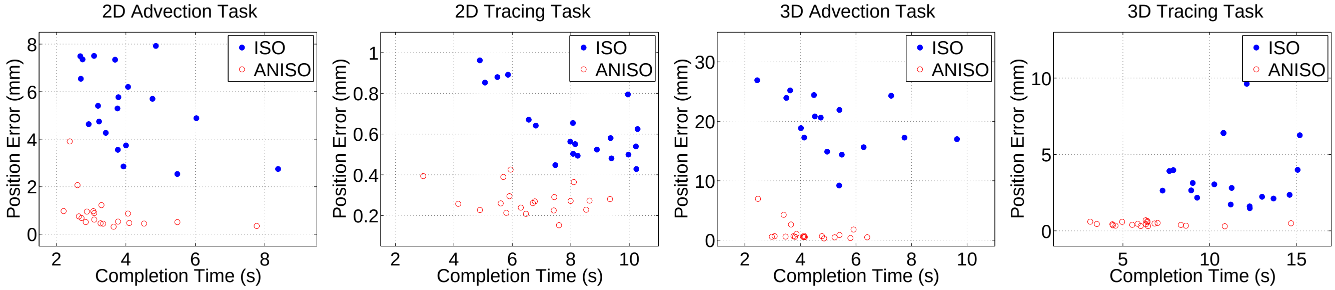


Figure 11: Distribution of position error versus completion time for the four tasks. The results show that shorter completion times correspond to larger errors for the ISO condition, but not necessarily for the ANISO condition.

[16] A. Mascarenhas, S. Ehmann, A. Gregory, M. Lin, and D. Manocha. Six degree-of-freedom haptic visualization. In *Proc. Touch in Virtual Environments*, Los Angeles, CA, Feb. 2001.

[17] E. Méndez, S. Yoshida, H. Noma, R.W. Lindeman, Y. Yanagida, S. Masaki, and K. Hosaka. A haptic-assisted guidance system for navigating volumetric data sets. In *Proc. World Haptics Conference*, Pisa, Italy, Mar. 2005.

[18] A. Nahvi and J.M. Hollerbach. Display of friction in virtual environments based on human finger pad characteristics. In *Proc. ASME Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 179–184, Anaheim, CA, Nov. 1998.

[19] M.A. Otaduy and M.C. Lin. User-centric viewpoint computation for haptic exploration and manipulation. In *Proc. IEEE Visualization*, pages 311–318, San Diego, CA, Oct. 2001.

[20] P. Passmore, C. Nielsen, W. Cosh, and A. Darzi. Effects of viewing and orientation on path following in a medical teleoperation task. In *Proc. IEEE Virtual Reality*, pages 209–215, Yokohama, Japan, Mar. 2001.

[21] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 2nd edition, 1992.

[22] 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, Dallas, TX, Nov. 1997.

[23] A. Seeger, A. Henderson, G.L. Pelli, M. Hollins, and R.M. Taylor II. Haptic display of multiple scalar fields on a surface. In *Proc. Workshop on New Paradigms in Information Visualization and Manipulation*, pages 33–38, Washington, D.C., Nov. 2000.

[24] T.V. Thompson II, D.E. Johnson, and E. Cohen. Direct haptic rendering of complex trimmed NURBS models. In *Proc. ASME Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, Nashville, TN, Nov. 1999.

[25] E. Vidholm and I. Nyström. A haptic interaction technique for volume images based on gradient diffusion. In *Proc. World Haptics Conference*, Pisa, Italy, Mar. 2005.

[26] 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, Aug. 1995.