

Manohar B. Srikanth · P. C. Mathias · Vijay Natarajan · Prakash Naidu · Timothy Poston

Visibility Volumes for Interactive Path Optimization

Abstract We describe a real-time system that supports design of optimal flight paths over terrains. These paths either maximize view coverage or minimize vehicle exposure to ground. A volume-rendered display of multi-viewpoint visibility and a haptic interface assists the user in selecting, assessing, and refining the computed flight path. We design a three-dimensional scalar field representing the visibility of a point above the terrain, describe an efficient algorithm to compute the visibility field, and develop visual and haptic schemes to interact with the visibility field. Given the origin and destination, the desired flight path is computed using an efficient simulation of an articulated rope under the influence of the visibility gradient. The simulation framework also accepts user input, via the haptic interface, thereby allowing manual refinement of the flight path.

Keywords Visibility analysis · flight paths · force-directed ropes · haptics · multimodal interaction

This work was done when the first author was with the Indian Institute of Science and the fourth author was with the Centre for Artificial Intelligence and Robotics, Bangalore, India.

Manohar B. Srikanth
Department of Mechanical Engineering, Massachusetts Institute of Technology, Cambridge, USA
E-mail: srimano@mit.edu

P. C. Mathias
NMR Research Centre, Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore 560012, India
E-mail: pcm@sif.iisc.ernet.in

Vijay Natarajan
Department of Computer Science and Automation, Supercomputer Education and Research Centre, Indian Institute of Science, Bangalore 560012, India
E-mail: vijayn@csa.iisc.ernet.in

Prakash Naidu
University of Toronto, Toronto, Canada
E-mail: crjp.naidu@utoronto.ca

Timothy Poston
National Institute for Advanced Studies, Indian Institute of Science Campus, Bangalore 560012, India

1 Introduction

Visibility is an important criterion for mission planning over terrains, both for civilian and military applications. It is vital to present a user with clear and complete information about the critical covert / exposed regions within a terrain for effective design and organization of a mission. A simple terrain visualization application that measures visibility of individual points on the terrain from an individual camera location leaves it hard to consider the entire terrain and the whole mission. For example, the visibility of air-borne vehicles to potential observers on the land is an important criterion and particularly crucial for surveillance and stealth navigation. Several applications including large areal illumination, rope-way design, cable layout design, antenna layout design, and swarm robotics have a similar requirement. We present a visualization and interaction system that enables mission planners to assess visibility over terrains and hence design an optimal flight path over the terrain.

Significant work has been carried out within the GIS and computational geometry communities on the development of sophisticated techniques for visibility analysis. Traditionally, this has meant the computation of visible portions of a terrain from a given point, or determining if an object is visible from a given view point. The answer to the former is a collection of regions, whereas the latter is a query resulting in a yes / no answer. We are interested in queries of the first kind. One challenge is to efficiently store the result of the query and visualize it in a useful manner. We address this challenge and develop a system that allows a user to design optimal flight paths. Several issues need to be addressed in the design of such a system:

- **Visibility measure:** What is a good measure of visibility? How do we compute the visibility measure efficiently?
- **Visualization:** What effective visualization techniques can help a user comprehend visibility information, and aid her in planning the mission?
- **Interaction:** What modalities of interaction best enhance visualization and make data exploration easy for the user?

- **Level-of-detail:** How do the various methods scale with resolution of the terrain and visibility range?

This paper’s contributions address all these issues. While visualization generally refers only to image synthesis, augmenting it with haptic cues (motion capture and force feedback) significantly enhances the information flow between human and the computer. We implement and demonstrate the benefit of multimodal interaction in our system.

Contributions. We develop a real-time system for visibility-guided interactive planning of optimal paths over terrains. Given a terrain, we generate a 3D scalar field defined on the volume above the terrain that measures visibility from various locations on the terrain. The large size of the terrain necessitates an efficient algorithm for computing this visibility field. We develop methods for optimal path computation over the terrain, path assessment, and haptic interaction. Our system also allows user-specified constraints and user-controlled path refinement. We compute a visibility force field from the visibility field, to guide the path optimization process. Paths are modeled as mechanical ropes subject to visibility forces. The user provides expert input through a haptic device and interactively plans a flight path. Figure 1 shows our visualization and interaction framework.

Section 2 presents prior work on visibility computation, haptics, and path optimization. Section 3 describes the visibility measure and its computation. Section 4 describes the use of visualization techniques to present the visibility information, optimal path computation, path assessment, and haptic interaction. Results of our experiments on terrains are presented in Section 5. Section 6 concludes the paper.

2 Related Work

Visualization of visibility information has been little addressed in the literature, except for trivial plotting and graphing techniques. However, several results are available on point-wise visibility computation, visibility analysis, and their application to GIS, robotic, and air-vehicle path planning applications. The surveys by Bittner and Wonka [3] and Cohen-Or et al. [5] provide a taxonomy of visibility problems and cover several applications. De Floriani and others have done significant work in the field of visibility analysis, with specific focus on efficient algorithms for terrain visibility [11, 12, 17]. Stewart [21] describes an efficient algorithm for computing the approximate horizon on terrains by dividing the terrain into sectors around the given point and considering samples within each sector. This algorithm is primarily used for shadow generation and illumination computation. Hence, visibility computation is restricted to viewpoints on the terrain. Our application, on the other hand, requires the computation of a visibility field over the entire volume above the terrain. Visibility algorithms depend on the nature of terrain encoding, popular ones being Regular Square Grid (RSG) and Triangulated Irregular Network (TIN). In the case of TINs, visibility computation involves finding the horizon and

building a horizon tree [6] to solve visibility queries for a point in logarithmic time. Building the horizon tree takes $O(N\alpha(N)\log N)$ time, where N is the number of number of sample points on the terrain. In the case of RSG representations, determining visibility is computationally expensive. Franklin [13] determines the visibility of a point by interpolating the slopes of nearby points in $O(N)$ time. Teng and Davis [22] discuss the same approach for parallel computation. A similar incremental wave-like approach for visibility calculation is adopted by Ben-Moshe et. al. [2] for TINs, however they consider only a sample set of points. Teng et al. [23,24] compute single point visibility on a hypercube machine with N processors in $O(\log N)$ time. Mills and Fox [16] also present solutions to intervisibility problems on fast parallel machines. Our approach is unique in that we construct an intermediate data structure once and use it for two purposes: interactive visualization of visibility and dynamic path planning.

Covert path planning generates paths that are hidden, to the extent possible, from observers on the ground. Marzouqi and Jarvis [15] propose an approach for generating paths that are sensitive to visibility from observers in an indoor environment with vertical walls and obstacles. Their approach is based on a distance transform, a map that supplies each point of the image with a distance to the nearest obstacle point. It uses a cost function to determine the path. However, they do not address the benefits of introducing visual presentations to aid a human in the loop for mission planning. Moreover, their technique is primarily useful for motion on a planar surface.

Virtual-reality interfaces offer several advantages for scientific visualization [4,7] Multidimensional scientific data are visualized using flexible virtual objects that provide an abstract representation of the locations of equilibria. Direct manipulation of the virtual objects allows intuitive exploration of the data and facilitates the discovery of features, which would be otherwise difficult to find via conventional visualization methods. In order to implement an effective multimodal scientific visualization application in virtual environment, issues of responsiveness and fast updates must be addressed. Haptic feedback is a promising interaction modality for a variety of applications [20]. The primary benefit of haptics is increased realism of simulation and improved operator performance. When presented with a proper combination of visual and haptic information, the visual cues are reinforced by the force feedback thereby making the interaction more realistic [10]. Some of the challenges for making interaction comfortable, fast, and effective are described in [8]. A majority of the previous methods for haptic displays of volume data properties are based on a fundamental relationship between the haptic probe position/velocity and the data at that point [1,27,14]. Algorithms for constrained point-based 3DoF haptic rendering have been developed for scalar density data [9,18]. Haptic stability is an issue in constrained haptic interaction where the haptic probe makes a rigid contact with the surfaces [25].

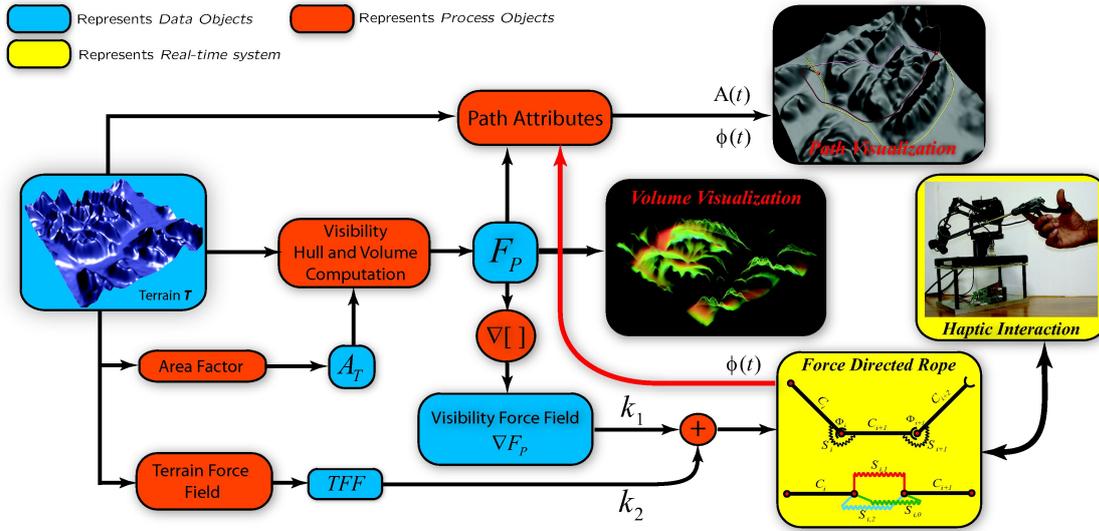


Fig. 1 Visualization pipeline: Given a terrain height field T , we compute the visibility field F_p once. We use standard volume rendering techniques for real-time visualization of F_p . The rope simulation component is a vital component of the systems, where the path is optimized based on gradient of F_p , called the visibility force field. Haptic device is used to interact with the rope and thus allows the user to explore the multiple equilibria. To avoid penetration of the rope with the terrain, we compute a terrain force field, which repels paths away from the terrain, and augment it with the visibility force field.

Our system uses rope simulation for interactive path optimization. The mechanics of rope and threads is hard to simulate at real-time rates; particularly for force feedback, which must be much faster than visual. Our rope simulation is based on multi-DoF articulated rigid body dynamics, which gives frame rates close to haptic requirements and restricts the rope to be inextensible.

3 Visibility measure and its computation

The input to our system is a terrain height field, defined as a bivariate function $\mathcal{T}(x, y)$ over domain D in \mathbb{R}^2 . A point P , with coordinates (P_x, P_y, P_z) , is called a *candidate point* if $P_z \geq \mathcal{T}(x, y)$. Two candidate points P, Q are said to be *mutually visible* if for all points R in the interior of the line segment PQ , $R_z > \mathcal{T}(R_x, R_y)$. Define a binary function

$$V(P, Q) = \begin{cases} 1 & \text{if } P \text{ visible to } Q \\ 0 & \text{otherwise} \end{cases}$$

Efficient computation of $V(P, Q)$ requires an intermediate data structure that enables fast queries. For a given candidate point P , we define a binary bivariate function

$$\mathcal{S}_P(x, y) = V(P, [x, y, \mathcal{T}(x, y)])$$

that identifies regions visible from P . The discrete version of \mathcal{S}_P is called a *shadowgram*. A radius of influence limits the range of visibility to a specified distance from P . The visibility of P is measured as the total terrain area visible from P assuming equal likelihood of presence of observers anywhere on the terrain. We introduce a real-valued weight

function $\mathcal{W}(x, y)$ to incorporate the higher / lower likelihood of observers at certain locations, and hence we define our visibility field \mathcal{F}_p at a point P as

$$\mathcal{F}_p(P) = \frac{\iint_{(x,y) \in D} \mathcal{S}_P(x, y) \mathcal{A}_{\mathcal{T}}(x, y) \mathcal{W}(x, y) dx dy}{\iint_{(x,y) \in D} \mathcal{A}_{\mathcal{T}}(x, y) dx dy}$$

$$\mathcal{A}_{\mathcal{T}}(x, y) = \sqrt{1 + \left(\frac{\partial \mathcal{T}}{\partial x}\right)^2 + \left(\frac{\partial \mathcal{T}}{\partial y}\right)^2}$$

where $\mathcal{A}_{\mathcal{T}}(x, y)$ is an area element at point (x, y) in the domain D .

The trivariate scalar field \mathcal{F}_p measures the visibility of candidate points from the terrain. It is important to note that it monotonically increases along positive z -axis. We use this property to design an efficient algorithm for computing it.

We define the measure of visibility \mathcal{F}_c for a parametric space curve ϕ as the integral of \mathcal{F}_p along ϕ

$$\mathcal{F}_c(\phi) = \int_0^1 \mathcal{F}_p(\phi(t)) \omega(t) dt$$

where ω is an importance function defined on the curve that captures user specified information along the curve. Given two curves, \mathcal{F}_c allows us to compare their relative visibility and hence to identify the better path.

While designing a path ϕ between points, we aim to minimize or maximize the functional \mathcal{F}_c with respect to ϕ . For example, minimization is appropriate for stealth navigation and maximization is appropriate for surveillance applications. Other optimization constraints can be incorporated including total length of curve, curvature, height from ground, clearance from terrain surface, etc.

Discrete visibility measures. We consider a *Regular Square Grid* (RSG) defined as a regular grid inducing a partition of the domain D into equally-sized rectangles. Let T be the discrete analog of \mathcal{T} , i.e., $T(i, j)$ denotes the terrain height for a grid point.

$$F_p[P] = \frac{\sum_i \sum_j S_P[i, j] A_T[i, j] W[i, j]}{\sum_i \sum_j A_T[i, j]}$$

where S_P, A_T , and W are discrete analogs of $\mathcal{S}_P, \mathcal{A}_T$, and \mathcal{W} , respectively. F_p can be computed and stored in a regular voxel grid. The discrete analog of \mathcal{F}_c is

$$F_c[\phi] = \sum_i F_p[\phi[i]] \omega(i).$$

Visibility computation. Computing F_p exactly could be time consuming: a naive computation for a $256 \times 256 \times 256$ voxel grid using standard ray-tracers takes approximately 100 days on a dual Xeon workstation! We will describe an algorithm for fast computation of an approximation of F_p .

Given a terrain point P , define the *visibility hull* (VH_P) as a surface that separates space into two regions such that all points above VH_P are visible from P and points below are not. It is constructed using Algorithm 1, an incremental procedure that computes the height of the visibility hull within rings of increasing size centered at P . The visibility hull for the first ring is identically equal to the terrain. Within each ring, an interpolation stencil specifies the rule to extend the existing hull. The extended hull is checked for intersection with the terrain height field and the slopes at the ring is updated accordingly. The value of F_p at voxels that intersect the visibility hull are incremented by a value equal to the area factor at P . Figure 2 illustrates the visibility hull computation procedure. This incremental construction of the visibility hull leads to an efficient computation of the visibility field.

To compute and store the visibility field, we create a voxel grid representing the volume above the domain D . A voxel stores a scalar value, initialized to zero, representing the visibility F_p at that point. The contribution of a terrain point $P = T[i, j]$ to the visibility field is computed as follows: Compute the visibility hull VH_P and accumulate area factor $A[i, j]$ to scalar value stored at voxels that intersect VH_P . The value of F_p at a voxel is now given by the sum of scalar values stored at all voxels below it. So, F_p can be computed now in a single sweep over voxels from bottom to top.

We introduce a notion of limited visibility by defining a cube of influence. For a point P on the terrain, define an axis aligned cube C with sides $2r$ such that P is the midpoint of the bottom face of C . Visibility is now defined as

$$V^C(P, Q) = \begin{cases} V(P, Q) & \text{if } Q \text{ is inside } C \\ 0 & \text{otherwise} \end{cases}$$

Algorithm 1 Visibility Hull Computation

- 1: **Input:** Terrain T , ring buffer RB to store line slopes with respect to the XY plane, area factor A , point P
 - 2: **Output:** Updated F_p for voxels intersecting the visibility hull VH_P
 - 3: Initialize RB to zero
 - 4: Set slope of all *cells* in first *ring* equal to slope of line joining P with mid-point of the cell
 - 5: **for** each *ring* **do**
 - 6: **for** each *cell* in *ring* **do**
 - 7: Determine slope of *cell* using interpolation stencil
 - 8: $L \leftarrow$ line with slope equal to $RB[cell_x, cell_y]$ and whose orthogonal projection onto the XY plane is the line joining $[P_x, P_y, 0]$ and $[cell_x, cell_y, 0]$
 - 9: Determine height H of L by computing voxel $[cell_x, cell_y, H]$ that intersects L
 - 10: **if** $H > T[cell_x, cell_y]$ **then**
 - 11: Increment $F_p[cell_x, cell_y, H]$ by A
 - 12: **else**
 - 13: $Q \leftarrow [cell_x, cell_y, T[cell_x, cell_y]]$
 - 14: $RB[cell_x, cell_y] \leftarrow slope(PQ)$
 - 15: Increment $F_p[Q]$ by A
 - 16: **end if**
 - 17: **end for**
 - 18: **end for**
-

The size of C determines the extent of visibility of a point and could vary spatially over the terrain. For the purpose of optimizing paths and for effective visualization, we require a level-of-detail representation for F_p obtained using different resolutions of the terrain and sizes for C . A larger C captures global visibility smearing out the local effects, while a smaller C takes local visibility into account. We consider the terrain at coarser resolution when C is large and at higher resolution for smaller C . A small cube could adversely affect the path optimizer because of possibly small variation in visibility within the extent of C . So, it is important to find a good balance between cube size and terrain resolution.

4 Visibility-aware paths

Visualization. The visibility field is visualized using existing volume rendering techniques where the user designs a transfer function to highlight specific regions of interest. Appropriate choice of a transfer function is crucial for effective visualization. We also implement a dynamic transfer function that changes smoothly over time thereby providing temporal cues to the variation of visibility in the region of interest. Figure 3 shows a volume rendered image for a synthetically generated terrain.

Path planning refers to generating paths, flight paths in our case, that obey kinematic constraints of the mobile object and also the environmental constraints such as obstacle avoidance etc. Abundant literature is available for path generation between given two points in space that avoid obstacles while satisfying kinematic constraints. We consider the problem of computing paths between a give pair of points that optimizes visibility over the path. Our approach is based on the simulation of a novel *force directed rope* (FDR) for

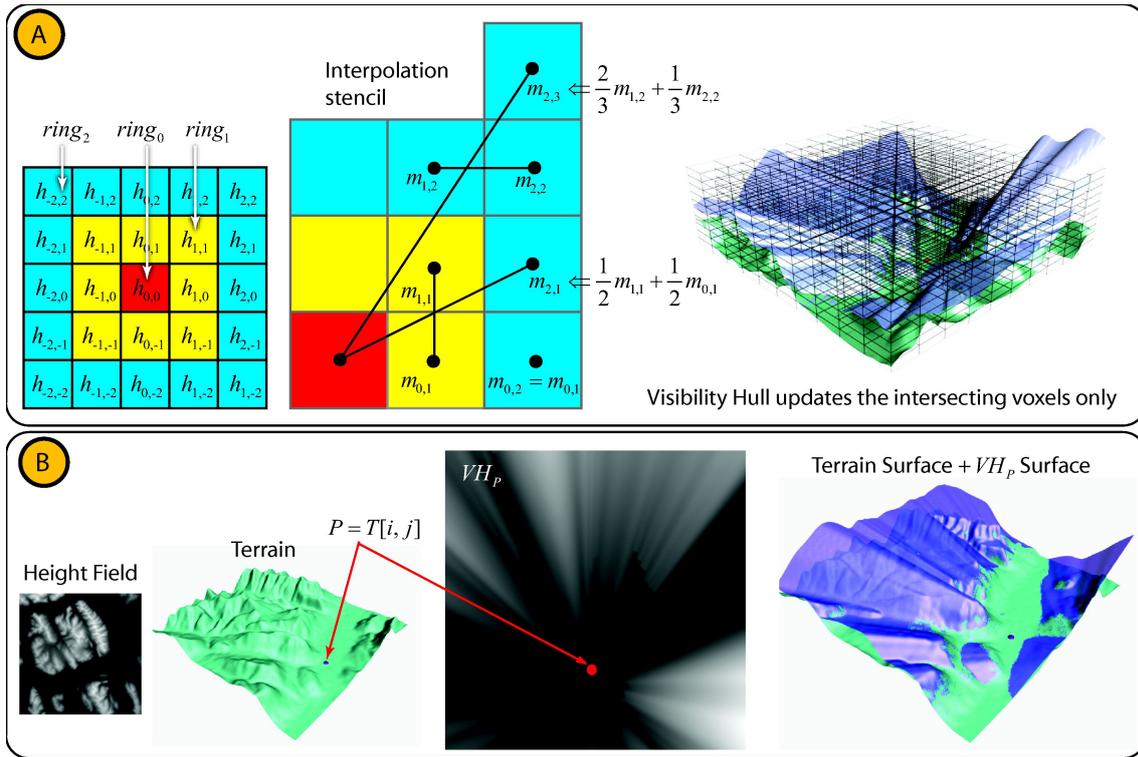


Fig. 2 Visibility Hulls. (A) The visibility hull of $P = h_{0,0}$ is computed at cells within rings of increasing size. An interpolation stencil specifies the rule to extend the hull to subsequent rings. For example, the height of the hull cell (2,3) is computed as $\frac{2}{3}m_{1,2} + \frac{1}{3}m_{2,2}$. (B) Various views of the visibility hull VH_P . All points above VH_P are visible from P .

interactive path optimization. Define *visibility force* as the gradient of F_p that acts upon a FDR. In its equilibrium state, the FDR optimizes F_c . Since F_p is discontinuous on the terrain surface, we compute the gradient ∇F_p for points above the terrain surface using backward differences.

$$\nabla F_p = \begin{cases} \left(\frac{\partial F_p}{\partial x}, \frac{\partial F_p}{\partial y}, \frac{\partial F_p}{\partial z} \right) & z > T(x,y) \\ (0, 0, 0) & \text{otherwise} \end{cases}$$

Interaction is key in mission planning. A user-assisted interactive approach significantly enhances the temporal visualization necessary for effective path planning. The rope dynamics gives the user a visual cue to visibility along the path. Force feedback provides the user with haptic cues of visibility on the whole path, specifically at the grasped point.

Force directed rope. A force directed rope is a mechanically simulated rope that responds to user interaction similar to a normal rope, but is subjected to various constraints besides its internal forces. We now describe two implementations of the FDR. The first technique models the FDR as a spring-mass system (Figure 4) where the rope is decimated to mass points and springs between them that impose constraints on the motion of the mass elements. The compliant nature of this model allows the rope to extend to minimize or maximize visibility. The spring's stiffness parameter controls the extent to which the rope can be extended. The sec-

ond implementation uses rigid capsules connected by ball-socket joints to model inextensible rope. The choice of a particular type of rope model is decided by the mission type (surveillance or stealth) and the type of constraints imposed.

Spring-mass system. Let $X_i(n)$ and $V_i(n)$ be the position and velocity of particle i with mass m at instance n . The force F_i applied on i would move it to a new position $X_i(n+1)$ and acquire a new velocity. We compute these using a modified Verlet integration [26]

$$X_i(n+1) = (2-d)X_i(n) - (1-d)X_i(n-1) + F_i(n) \frac{\Delta t^2}{m},$$

$$V_i(n+1) = V_i(n) + \frac{F_i(n) + F_i(n+1)}{2} \frac{\Delta t}{m},$$

where $0 \leq d \leq 1$ is the damping factor and Δt ($= 1ms$) denotes the time step. $F_i(n)$ is the sum of the internal spring forces and the external force field described in Equation 1. Spring S_i applies a force

$$F_{i,i+1}^S = \frac{K_S(\|X_i - X_{i+1}\| - L_{i,i+1})(X_i - X_{i+1})}{\|X_i - X_{i+1}\|} + K_V(V_i - V_{i+1}),$$

on mass points m_i and m_{i+1} , where $L_{i,i+1}$ is the rest length, K_S is spring stiffness, and K_V is the damping constant of the spring.

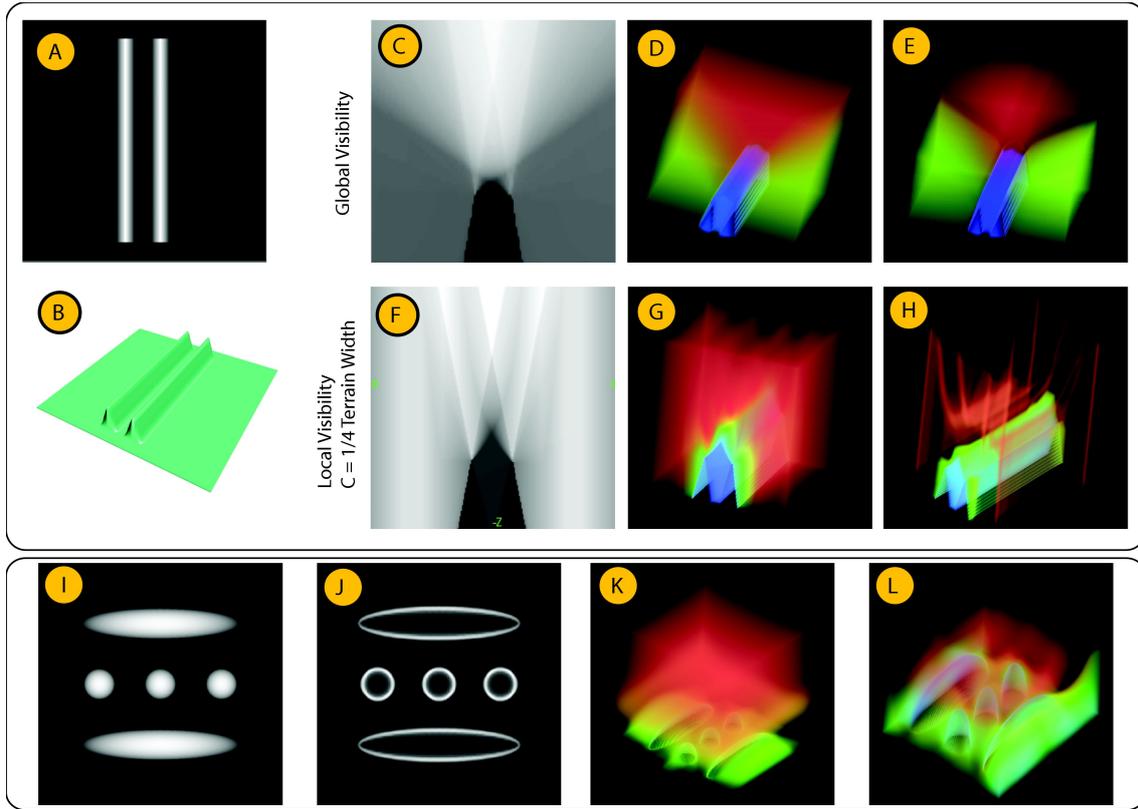


Fig. 3 Volume rendered visibility field (Images were rendered using VolView from Kitware). (A) Grayscale colormapped image of a synthetically generated terrain. (B) Surface rendering of the terrain. (C-E) A vertical slice of the visibility field and volume rendered images of the visibility field. (F-H) Vertical slice and volume rendered images of visibility field computed using a smaller cube of influence. (I) Grayscale colormapped image of a terrain with two extended peaks and three sharp peaks. (J) Area factors of the terrain, (K,L) Volume rendered images of the visibility field.

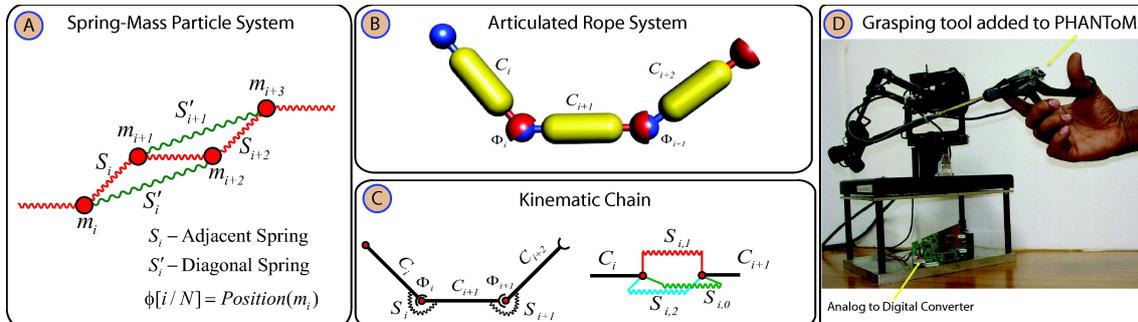


Fig. 4 Force Directed Rope: (A) A simple spring-mass system: a chain of particles, each particle demonstrating 3 dof motion. Diagonal springs resist bending. (B,C) Articulated rope consisting of rigid capsules connected by spring loaded ball-socket joints. (D) Sensable Phantom Premium haptic device used in our system. A grasping holder was developed and attached to the Phantom device resulting in an intuitive interface for rope manipulation.

The spring model is simple and fast, but it may introduce aliasing artifacts such as jagged edges because forces are sampled only at mass points and not on the piecewise-linear path. Therefore, a large number of mass points are required for numerical convergence.

Articulated Rope. The high resolution requirement in spring-mass systems can be overcome if the rope is modeled as

an articulated rigid body [19]. N rigid capsules $\{C_i\}_{i=0}^{N-1}$ are connected end-to-end by spherical joints $\{\Phi_i\}_{i=0}^{N-2}$ to impose constraints on their motion, see Figure 4. Joint Φ_i connects C_i with C_{i+1} . Each joint Φ_i has three springs $\{S_{i,j}\}_{j=0}^2$ that model resistance to bending and axial twisting. The motion of the capsules obeys six degree-of-freedom linear and rotational dynamics. The haptic proxy point is coupled to the grasped point on the rope via a soft-joint which allows for

compliance and keeps the rope from getting stretched beyond limits. All simulation events are coordinated using a timer. The position X_i and orientation q_i of C_i are updated using Euler time integration.

$$\begin{aligned} X_i[n+1] &= X_i[n] + \Delta t \cdot V_i[n] \\ q_i[n+1] &= \hat{q}_i(\Delta t \cdot \omega_i[n]) \cdot q_i[n] \end{aligned}$$

where $\hat{q}(\omega) = [\cos(|\omega|/2), \sin(|\omega|/2) \cdot \omega/|\omega|]$ is a unit quaternion. The velocity and angular momentum are updated using forward Euler time integration.

Force Field. The capsules exhibit linear and rotational motion under the influence of a force field

$$G = k_1 \nabla F_p + k_2 TFF + k_3 U_1 + k_4 U_2 + \dots, \quad (1)$$

where ∇F_p is the gradient of the visibility field, TFF is the *terrain force field* that keeps the FDR away from the surface of the terrain, and U_1, U_2, \dots are additional user forces. The weights k_1, k_2, \dots control the influence of each force on the rope and their signs control the direction of motion of the elements of the FDR. The linear component of the force acting on C_i is computed as the average over the length of the capsule, and the rotational force (torque) is computed about the center of gravity of C_i . Both components are used to compute the new position and orientation of the capsules. The spherical joint maintains the connectivity between capsules and the springs resist bending thereby producing smoother ropes. Inextensibility of this rope model is useful when there is a constraint on the total length of the path. This model is superior to the spring-mass model in that it does not produce aliasing artifacts.

Terrain force field. The primary force acting on the rope is

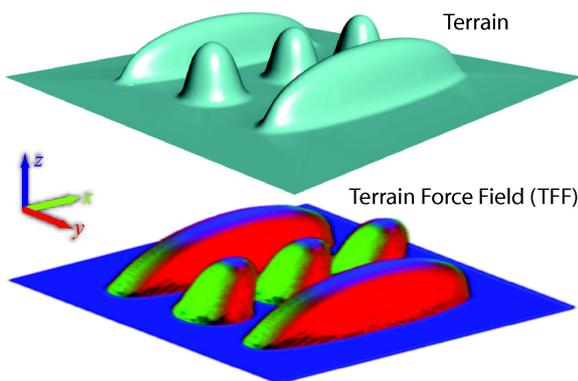


Fig. 5 Terrain force field TFF is computed around the terrain up to a constant offset from the surface in both directions. TFF repels the rope away from the surface of the terrain.

due to the visibility field. To prevent the rope from penetrating the terrain during free re-configuration (without haptic intervention) we need to detect collisions between the FDR

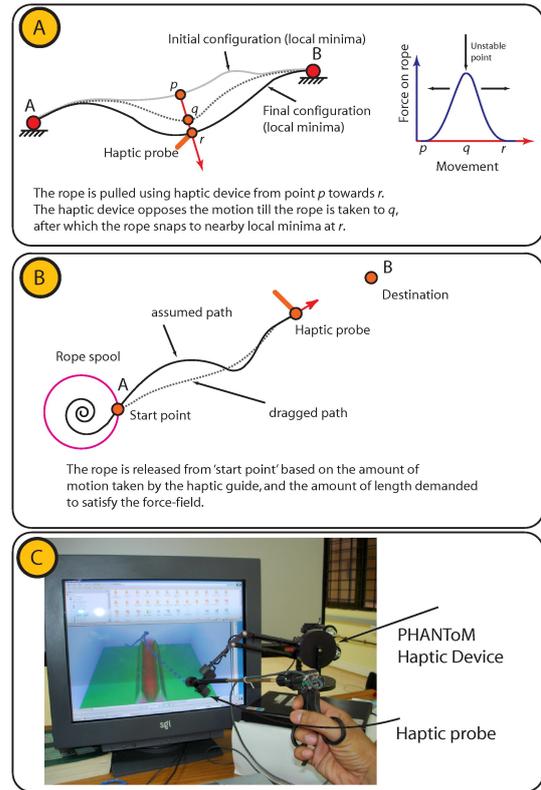


Fig. 6 Haptic interaction: (A) In the first interaction scheme, the rope is fixed at the end points, the user grasps the rope and displaces it. (B) In the second scheme, the rope is spooled at one end and the user drags the other end toward the destination along a desired path. (C) The haptic device in action.

and the terrain surface. Instead of detecting the collisions explicitly, we design a terrain force field that repels the FDR away from the terrain.

To compute the repulsion force at a space point, we locate the nearest terrain point. The line joining a point in space with its closest terrain neighbor aligns with the normal to the terrain at the terrain neighbor. Therefore, in order to identify nearest terrain neighbors for all space points, we compute the intersections between offsets of the triangulated terrain surface and voxels in the volume above. All voxels that intersect a triangle of the terrain surface store the closest point on that triangle in the first step. Next, the triangle mesh is offset in the direction of the vertex normals by a small distance followed by the intersection test. If the intersection test reveals a closer point, the voxel is updated appropriately. We repeat the above step until all voxels in the neighborhood of the terrain are traversed. Figure 5 shows the TFF computed using our method. Though computing TFF is time consuming, it is done only once for a terrain. This is particularly advantageous for real-time rope dynamics. TFF computation can be replaced by a terrain collision detection and response system, which will be too slow for real-time rope dynamics.

Haptics. A kinesthetic 3 dof haptic device is used to interact with the FDR. The user modifies the FDR configuration by grasping a point on the FDR and moving it to the desired location. The forces felt in the haptic device indicates visibility over the path. A strong force indicates motion away from a desired location (local minima / maxima). The haptic device has two modes of interaction with the rope. In the first mode, the rope is fixed at the two end points, the user grabs and displaces the rope thereby physically exploring multiple equilibria. In the second mode of interaction, the rope is spooled at one end and the other end is dragged in the desired path, towards the destination using the haptic device. During this process, the rope reconfigures to optimize for visibility (see Figure 6).

5 Experiments

We performed our experiments on both synthetically generated terrains and terrains obtained from United States Geological Survey (USGS). Experiments on the synthetic terrain allowed us to test the algorithm in a controlled and well understood environment. Experiments on a real terrain allowed us to study the effectiveness of our method. The synthetically generated terrain consisted of two parallel prisms located along the length and near the center of a square flat area (see Figure 7). Given various initial configurations of source and destination points, the objective was to maximize visibility. In each case, the rope simulation quickly and correctly identified the desired path. The visibility maximizing path always passed over the valley between the prisms.

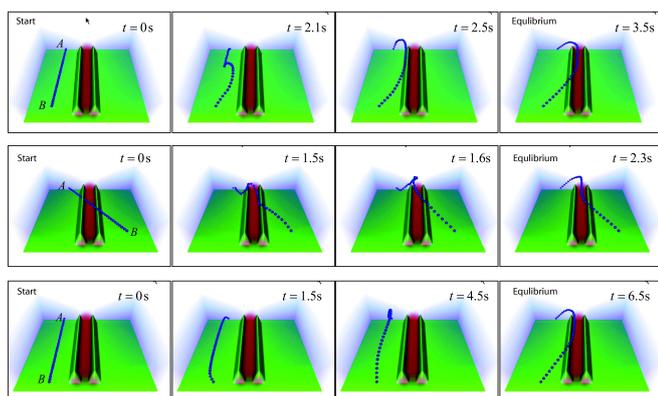


Fig. 7 Rope simulation with k_1 set to 1 for maximizing visibility over a synthetic terrain. The visibility field is shown as a volume rendering with red indicating low values and blue indicating high values. A and B are the source and destination points, respectively. Top row: A and B are located 10 units above ground. The rope reaches the equilibrium state in 3.5s. Middle row: The destination is moved to lie diagonally across the terrain. Bottom row: A and B are placed similar to the first row but 1 unit above ground. The FDR reaches equilibrium after a longer period of time. However, in all three cases, the flight path attempts to pass above the valley in order to maximize visibility.

The terrain from USGS is available as a DEM, which we sampled at 512×512 resolution. We computed the area factor and the visibility field over a $512 \times 512 \times 128$ voxel grid. For this experiment, we worked with two users from the Institute for Robotics and Intelligent systems, Centre for Artificial Intelligence and Robotics, Bangalore: a mission planner who was the primary user and a user-interface expert who helped evaluate the usability of the haptic interface for rope manipulation. In the following, we discuss the experiment and collective observations of the two users.

The experiment was performed for a terrain region in the East Bay Hills of California near San Francisco, see Figures 8 and 9. We briefed the user about the system including a description of the terrain, visibility hull, visibility field, and haptic interface. The task was to find a path that minimizes / maximizes visibility while satisfying additional user-imposed criteria. The user traced initial paths between two pairs of points, Q to P and Q to R , using a 2D mouse-based entry tool, see Figure 10. Our path tracing tool created a space path from the 2D path by lifting it appropriately above the terrain. Next, the user with the help of the volume visualization tool studied the distribution of the visibility field over the entire terrain and specifically near the region between source and destination.

Figure 10 shows the result of the experiment. The rope simulation computed an optimal path given the user-traced space path as an initial configuration. The user controlled the values of parameters k_i to increase / decrease the relative influence of the visibility field. Negative values of k_1 helped minimize the visibility objective, F_c . The resulting

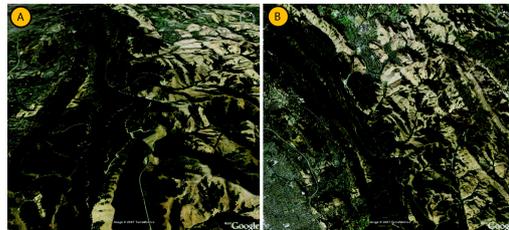


Fig. 8 Path planning experiments were performed on terrain region in the East Bay hills of California, between Oakland and Moraga. Coordinates: 3746019:92" North, 12206016:72" West. Image taken from Google EarthTM (earth.google.com). (A) Oblique view showing valleys and peaks. (B) Orthographic view from top.

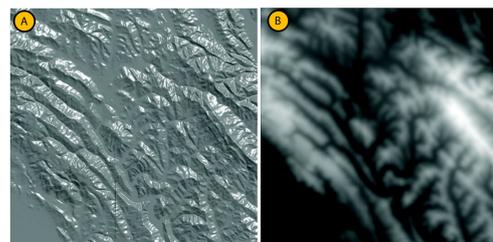


Fig. 9 (A) Surface rendering and (B) raw height field of the location shown in Figure 8. The digital elevation model (DEM) is available from US Geological Survey (<http://www.usgs.gov>)

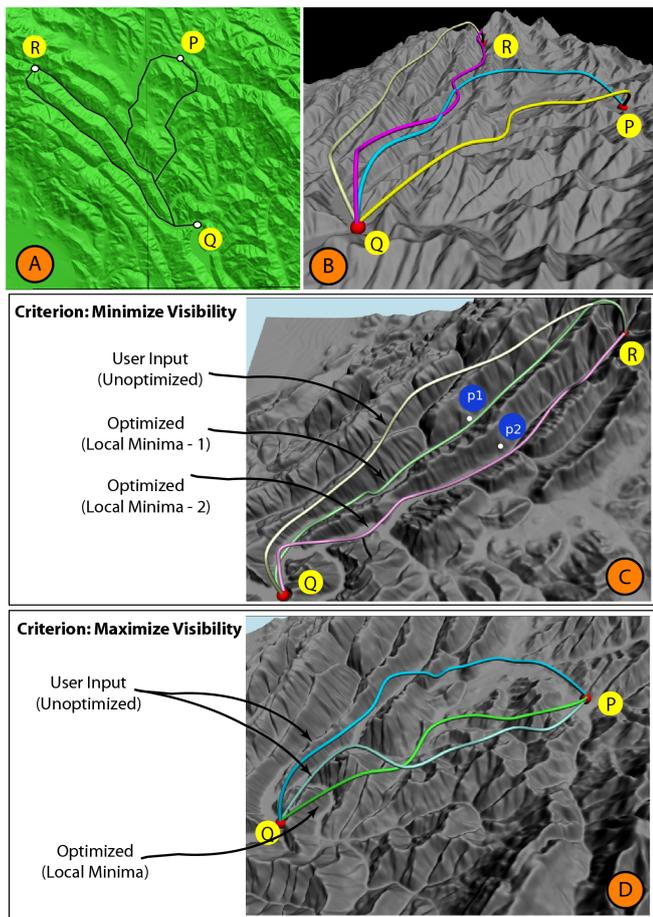


Fig. 10 Mission planning experiment over the East Bay hills. (A) User supplied a set of initial paths from Q to P and R . (B) The 2D paths were lifted above the terrain and the resulting space paths were fed to the rope simulation as initial configurations. (C) The FDR identified a minimum visibility (green) path between Q and R . The user manipulated the rope using the haptic device to incorporate additional constraints, namely p_1 should be hidden and p_2 should be visible from the path. The rope simulation included these constraints to find a new optimal (purple) path. (D) Given a visibility maximization criterion, the FDR converged to the same path between Q and R from two different initial configurations.

path passed through a deep valley and stayed as close to the terrain as possible. The user then identified a point p_1 that should necessarily remain hidden from the path and a second point p_2 that should necessarily be visible from the path. The haptic interface allowed the user to manipulate the rope by pulling it away from p_1 and towards p_2 . While manipulating the rope to satisfy the two additional constraints, the user tuned the rope stiffness parameter, which essentially controls the flexibility of the rope. This flexibility allowed the user to impose additional constraints with ease. In addition to the haptic feedback, the user found the visibility hull to be of great help as a visual aid while specifying the additional constraints. The visibility field was rendered with higher transparency values during the rope simulation phase in order to help the user focus his attention on the

rope. Subsequent to the rope manipulation, the rope simulator computed the appropriate visibility minimizing path, which now passed a different valley but remained close to the terrain. We repeated the entire experiment for maximizing visibility and obtained the results shown in Figure 10-D. The user found our visibility-field-driven, haptic-assisted, user-in-loop approach to be intuitive and efficient for generating optimal paths when compared to manual path planning.

The visibility computation took 93 seconds and the haptic loop executed at a frequency of 1000 Hz. The path reached the equilibrium state within a span of 3 – 4 seconds, depending on the extent to which it was perturbed. An Intel dual Xeon workstation running at 3.0 GHz was used for the computation and user interaction. The volume was visualized using a second workstation.

6 Conclusions and Future Work

We have described an interactive system for generation, assessment, and interactive planning of flight paths that optimizes a given visibility criterion. Efficient pre-computation of a visibility field and efficient simulation of an articulated rope simulation results in interactive computation of optimal paths between two end points specified by a user. The system allows the user to specify additional constraints on the path using both visual and haptic feedback. The seamless integration between the interaction mechanism and the path optimizer results in a very effective system. The development of a mission planning system that can accept several complex optimality criterion while providing interactive optimization capability remains to be a challenging problem.

Acknowledgements This work was done in collaboration with Institute for Robotics and Intelligent systems, Centre for Artificial Intelligence and Robotics (CAIR), Defense Research and Development Organization (DRDO), India. The authors thank the Virtual Reality and Robotics group of CAIR for providing support and feedback on the visibility measures. The authors also thank the Supercomputer Education and Research Centre (SERC) of Indian Institute of Science for providing the computational facilities. Thanks to Subrata Rakshit, Sitaram N., and V. S. Mahalingam for providing valuable feedback on initial proposals.

References

1. Avila, R.S., Sobierajski, L.M.: A haptic interaction method for volume visualization. In: Proc. IEEE Visualization, pp. 197–204 (1996)
2. Ben-Moshe, B., Carmi, P., Katz, M.J.: Approximating the visible region of a point on a terrain. In: L. Arge, G.F. Italiano, R. Sedgewick (eds.) Proc. Sixth Workshop on Algorithm Engineering and Experiments and the First Workshop on Analytic Algorithms and Combinatorics (ALENEX-04), pp. 120–128 (2004)
3. Bittner, J., Wonka, P.: Visibility in computer graphics. Journal of Environment and Planning B: Planning and Desig 5(30), 729–756 (2003)

4. Bryson, S.: Virtual reality in scientific visualization. *Communications of the ACM* **39**(5), 62–71 (1996)
5. Cohen-Or, D., Chrysanthou, Y., Silva, C., Durand, F.: A survey of visibility for walkthrough applications. *IEEE Transactions on Visualization and Computer Graphics* **9**(3), 412–431 (2003)
6. Cole, R., Sharir, M.: Visibility problems for polyhedral terrains. *J. Symbolic Computation* **7**(1), 11–30 (1989)
7. Cruz-Neira, C., Leigh, J., Barnes, C., Choen, S., Das, S., Englemann, R., Hudson, R., Papka, M., Siegel, L., Vasilakis, C., Sandin, D.J., Defenti, T.A.: Scientists in wonderland: a report on visualization applications in the CAVE virtual reality environment. *Proc. IEEE Symp. Research Frontiers in Virtual Reality* pp. 59–67 (1993)
8. van Dam, A., Forsberg, A.S., Laidlaw, D.H., Jr., J.J.L., Simpson, R.M.: Immersive VR for scientific visualization: a progress report. *IEEE Computer Graphics and Applications* **20**(6), 26–52 (2000)
9. Donald, B.R., Henle, F.: Using haptic vector fields for animation motion control. In: *Intl. Conf. Robotics and Automation*, pp. 3435–3442 (2000)
10. Durlach, N.I., Mavor, A.S. (eds.): *Virtual Reality: Scientific and Technological Challenges*. National Academy Press, Washington, D.C. (1995)
11. Floriani, L.D., Magillo, P.: Algorithms for visibility computation on digital terrain models (1996)
12. Floriani, L.D., Magillo, P.: Intervisibility on terrains. In: P.A. Longley, M.F. Goodchild, D.J. Maguire, D.W. Rhind (eds.) *Geographic Information Systems: Principles, Techniques, Management and Applications*, pp. 543–556. John Wiley & sons (1999)
13. Franklin, W.R., Ray, C.K.: Higher isn't necessarily better: visibility algorithms and experiments. In: *Sixth International Symposium on Spatial Data Handling*, vol. 2, pp. 751–763 (1994)
14. Lawrence, D.A., Pao, L.Y., Lee, C.D., Novoselov, R.Y.: Synergistic visual/haptic rendering modes for scientific visualization. *IEEE Computer Graphics and Applications* **24**(6), 22–30 (2004)
15. Marzouqi, M.S., Jarvis, R.A.: New visibility-based path-planning approach for covert robotic navigation. In: *Robotica*, vol. 24, pp. 759–773. Springer (2006)
16. Mills, K., Fox, G., Heimbach, R.: Implementing an intervisibility analysis model on a parallel computing system. *Computers & Geosciences* **18**(8), 1047–1054 (1992)
17. Nagy, G.: Terrain visibility. *Computers & Graphics* **18**(6) (1994)
18. van Reimersdahl, T., Bley, F., Kuhlen, T., Bischof, C.H.: Haptic rendering techniques for the interactive exploration of CFD datasets in virtual environments. In: *EGVE '03: Proc. Workshop on Virtual environments*, pp. 241–246 (2003)
19. Srikanth, M.B., Mathias, P., Naidu, P., Poston, T., Ramachandra, R.: Real-time articulated rope simulation. *Journal of Virtual Reality* (2007). Under review
20. Srinivasan, M.A., Basdogan, C.: Haptics in virtual environments: taxonomy, research status, and challenges. *Computers & Graphics* **21**(4), 393–404 (1997)
21. Stewart, A.J.: Fast horizon computation at all points of a terrain with visibility and shading applications. *IEEE Transactions on Visualization and Computer Graphics* **4**(1), 82–93 (1998)
22. Teng, Y.A., Davis, L.S.: Visibility analysis on digital terrain models and its parallel implementation. Technical Report CAR-TR-625, Centre for Automation Research, University of Maryland, College Park, Maryland (1992)
23. Teng, Y.A., DeMenthon, D., Davis, L.S.: Stealth terrain navigation. *IEEE Transactions on Systems, Man, and Cybernetics* **23**(1), 96–110 (1993)
24. Teng, Y.A., Mount, D.M., Puppo, E., Davis, L.S.: Parallelizing and algorithm for visibility on polyhedral terrain. *Intl. J. Comput. Geometry Appl* **7**(1/2), 75–84 (1997)
25. Vasudevan, H., Srikanth, M.B., Manivannan, M.: Rendering stiffer walls: a hybrid haptic system using continuous and discrete time feedback. *Advanced Robotics* **21**(11), 1323–1338 (2007)
26. Verlet, L.: Computer 'experiments' on classical fluids. I. thermodynamical properties of lennard-jones molecules. *Phys. Rev.* **159**(1), 98 (1967)
27. Yano, H., Yoshie, M., Iwata, H.: Development of a non-grounded haptic interface using the gyro effect. In: *HAPTICS*, pp. 32–39. IEEE Computer Society (2003)



Manohar B. Srikanth is a graduate student and research assistant at MIT. He obtained his MS in Bangalore from the Indian Institute of Science in 2007, after running VR and machine interaction projects at the Defence Research and Development Organisation. He currently works in adaptive control and simulation for UAVs. He likes multidisciplinary work that include scientific visualization, haptics and control. He has several pending patents and published journals and conference papers.



P. C. Mathias is currently an associate professor in the NMR Research Center, Indian Institute of Science, Bangalore. His research and teaching interests include computer graphics and visualization, digital signal processors and NMR instrumentation.



Vijay Natarajan is an assistant professor in the Department of Computer Science and Automation and an associate faculty in the Supercomputer Education and Research Centre at the Indian Institute of Science, Bangalore. He received the Ph.D. degree in computer science from Duke University in 2004. He holds the B.E. degree in computer science and M.Sc. degree in mathematics from Birla Institute of Technology and Science, Pilani, India. His research interests include scientific visualization, computational geometry, computational topology, and meshing.



Prakash Naidu received his PhD degree in engineering from University of Toronto, Canada in 1997. He has worked in or has been affiliated with leading institutions, laboratories, and industry in USA, Canada, and India in different capacities. He has several awards and honors including Canadian Commonwealth Fellowship and Engineering Design Awards. He is the originator of the field of Design for Automation and his research interests include design theory and methodology, design of intelligent systems in the fields of medical devices, biotechnology, advanced manufacturing, mechatronics,

robotics, automation, and virtual reality. He is a member of the American Society of Mechanical Engineers, and licensed member of the Association of Professional Engineers of Ontario (P.Eng.).



Tim Poston has a Math PhD from Warwick, England, and has published with co-authors from archaeologists to brain surgeons. He is an inventor on ten issued patents, fifteen pending, which range from a helical surgical probe to flexible ways to find, collaborate on and work with documents. He is co-author of two texts, on General Relativity and on Catastrophe Theory, in print since the 1970s. He has worked in haptics, visualization, and the cognitive interaction of humans and machines.