ECAD - A Prototype Screen-based VR Solid Modeling Environment Incorporating Tangible Deformable Models

Robert Blanding[1] and George Turkiyyah[2]

[1]Intel Corporation, rob.l.blanding@intel.com
[2]University of Washington, george@ce.washington.edu

ABSTRACT

Two shortcomings of the current generation of computer-aided design (CAD) software are (1) segregating the tasks of modeling and simulation, and (2) casting inherently 3D tasks, such as modeling and manipulation, into a 2D environment. The former is due to the fact that modeling is generally an interactive process whereas simulation often must be conducted off-line due to the computational cost of traditional simulation methods. The latter arises largely from the relative newness of 3D interface devices and the lack of software that takes good advantage of them.
In this paper we present a screen-based VR modeling and simulation environment called ECAD (Enhanced CAD). ECAD is a proof-of-concept system that combines 6 DOF input, 3D visualization, haptic feedback, and a new real-time deformation simulation engine into a unified modeling and simulation environment. Real-time simulation of deformations provides the user with immediate tangible feedback regarding the behavior of the model, allowing for more rapid design iteration and encouraging the trial-and-error exploratory shape modification typical in the early conceptual phase of design. 3D interfaces to the modeling environment also improve the overall flow of information between the user and the computer model. Stereoscopic display, haptic feedback, and 6 DOF interface devices allow for a much more intuitive interaction with the model and improved task completion times. We describe the modeling environment and show demonstrative examples.

Keywords: Real-time Simulation; Haptic CAD; User Interfaces.

1. INTRODUCTION

As three dimensional solid modeling has become a standard tool for the mechanical engineer, application interface designers have been forced to address the problem of enabling engineers to create and interact with a three dimensional object using only the two dimensional tools provided by the common workstation. This has generally resulted in cumbersome interfaces, awkward modes of interaction, and systems with steep learning curves. More importantly perhaps, the current generation of engineering design tools is not well suited to the highly exploratory and rapidly evolving nature of conceptual design. Analysis and simulation are segregated from the interactive modeling tasks, and are relegated to a separate process that's not tightly integrated with the CAD interface. Experimenting with shape and topology design changes to improve the mechanical characteristics of a 3D model is not intuitive. Feedback to the designer regarding performance measures such as stiffness, and stiffness-sensitivity to shape changes is not provided through direct, physical/tangible interfaces. Such capabilities would have significant impact on encouraging exploratory model changes and would ultimately contribute to a more rapid design cycle.
Recent advances in display and input technologies have the potential to free designers from the constraints imposed on them by traditional two-dimensional interfaces, enabling more intuitive interaction with 3D models, and providing more effective physical-manipulation environments for virtual prototyping. New interface devices, such as the 6 DOF PHANToM haptic device and 6 DOF Spaceball-like devices, coupled with LCD shutter-glasses or similar 3D display technology greatly simplify many common modeling tasks and open the door for entirely new modes of interaction with finite element simulations.
Tasks even as simple as manipulating surface control-points that can be frustrating and counter-intuitive in a 2D environment become perfectly natural, with accurate depth perception, when performed using 6 DOF interface devices and stereographic displays. Moving a control point becomes simply a matter of grabbing the control point, dragging it into a new position in space and dropping it into place. Modeling tasks that require manipulating more than two degrees of freedom simultaneously benefit significantly from the additional sensing channels of these

enhanced interfaces. Tasks that require additional cues and feedback (e.g., relative positioning, mating, joining, constraining, etc.) are much enhanced with force-feedback devices that allow users to experience the model via forces felt through the user-device interface (the input stylus in the case of the PHANToM). By using this feedback, a variety of user interaction is enabled, such as allowing the user to feel virtual objects, explore surface continuity, and sense collisions between objects in a tactile manner. Forces are also useful for restricting the degrees of freedom available to the user in order to simplify certain tasks (such as 2D sketching) when a full six degrees of freedom interaction may be disorienting and detrimental to performance.

In a CAD environment that encourages physical interaction with touchable 3D models, it becomes quite compelling to perform finite element simulations so that forces are rendered with the haptic devices while deformations (and perhaps strains) are rendered visually in 3D stereo. In the context of conceptual design for example, the intuition obtained by "feeling" the stiffness of the model in various deformation modes encourages experimentation and explorations of shape modifications. The user can push, pull, shear, bend, squeeze, etc., the model and obtain direct feedback on its stiffness and deformation characteristics, and then use that information to modify the geometry and/or topology of the model essentially solving a shape optimization problem through direct physical manipulation.

Current CAD systems are capable of simulating many aspects of a physical model, but for real-time interaction they are generally restricted to rigid-body simulations such as mass properties, interference detection, and kinematics. Simulating deformations and internal stresses is a separate process that is one or two orders of magnitude slower than the update rate that is required by most haptic devices (100-1000 Hz range). There are two main obstacles that prevent simulation of deformations at interactive rates. First, finite-element analysis requires complex intermediate representations of the object to be generated, usually with extensive user intervention. It is not uncommon for the construction of these intermediate models to be just as time-consuming as building the geometric model itself. Second, the complexity of the models required to generate accurate results is high enough that the computational cost of evaluating the models can easily become prohibitive.

In this paper we describe ECAD (short for Enhanced CAD), a proof-of-concept CAD system that we have assembled to allow us to experiment with developing stereo- and haptic-enhanced features for next-generation CAD environments, and evaluate their effectiveness in modeling and interacting with geometric models. By combining 3D interface devices with a novel real-time deformation simulation and haptic feedback we allow CAD users to design, edit, and interact with models in a manner far more intuitive than currently available CAD packages. The contributions of the paper are twofold: (1) models and a real-time solution engine for imbuing CAD models with physical characteristics. The finite element model is based on the skeleton of the geometric model. The real-time deformation engine is specifically formulated to allow the use of fast updates and avoid the expensive refactoring of the stiffness matrix as the user interacts with and modifies the model, even when these modifications change the topology of the model. These fast updates are the key for real-time interaction; and (2) the integration of 3D interface components into a novel, intuitive, and highly usable 3D user interface for CAD. We believe that commercial systems can implement these kinds of interface concepts economically to facilitate shorter and more cost-effective design cycles.

2. RELATED WORK

Haptic and VR CAD Systems. Scores of exploratory VR and haptic systems have been developed for CAD applications over the last decade or so by researchers experimenting with this new generation of interface devices. Early systems included the 3DM system [4], the FreeForm package from Sensable, the TIME system [6], the 3-Draw system [12], and many others. There has also been considerable interest in using virtual reality to simulate assembly of mechanical parts [7]. More recent systems include, for example, the Virtual Designworks [8], the CAD/CAM system presented in [19], and the virtual sculpting system in [10]. Ye and Campbell [16] discuss the importance of VR-haptic interfaces for supporting conceptual design. Despite this body of exploratory research work in this area, commercial CAD hardware interfaces still consist almost exclusively of a mouse / tablet, keyboard, and monitor, probably because of lack of effective integration with existing CAD modeling tools and the lack of compelling read-time simulation engines. Some of the challenges in developing 3D CAD interfaces are described in [11].

More elaborate systems from a hardware interface viewpoint such as the Haptics Workstation from Immersion and various exoskeleton multifinger devices with more detailed tactile and force feedback [13], have been designed. However such devices are unlikely to be used in CAD systems, where cost and workflow integration are important adoption criteria.

Immersive systems are an alternative to the screen-based system presented here. Demonstration systems have been developed where users wear head-mounted displays and navigate an immersive world by walking around models and manipulating them using gestures such as pointing and grabbing. Such an environment is certainly useful for some

tasks, especially visualization of large models on a one-to-one scale, however the practical usefulness of an immersive system for day-to-day engineering design tasks is questionable. The hardware associated with such systems tends to be quite costly and uncomfortable for the user, and the software requires reinventing several well-established interaction metaphors.
A screen-based VR system has the advantage of allowing a user to sit at a conventional workstation and only requires the addition of a few relatively inexpensive and familiar/comfortable devices. This is our focus in this work where we present a system that transparently integrates new interface tools with existing interface metaphors.

Real-time Simulation of Deformable Models. Deformable solid models have been used in interactive computer modeling and animation for decades. Approaches based on physical or quasi-physical models that minimize an appropriate energy metric inspired by the mechanics of deformation of continua, have produced very compelling results and useful systems over the years. The literature is far too large to survey meaningfully here, but recent works include that of [18] who present a system that can handle large deformations of 2D curves by minimizing a quadric energy function, and transfers the deformation to curves on a 3D mesh for solid deformation; and [3] who describe an editing system allowing the user to specify intuitively a region of influence and corresponding boundary conditions, and deforms the boundary surface by minimizing an approximate shell strain energy. Approaches based on geometry-inspired heuristics for deformations have also produced intuitive systems for the interactive deformation of solid models [9,14]. Haptic interfaces to CAD systems have also relied on physically-based models to provide intuitive, force-felt, interfaces for producing large nonlinear deformations of models in real time [1]. All these approaches however preserve the topology of the 3D model. In this work we propose incremental model formulations that allow real-time interaction with reasonably sized models even in the presence of topological changes.

3. MODELING FOR REALTIME INTERACTION

An effective haptic simulation must supply force data to the haptic devices at very high rates. The servo loop of a typical haptic device needs to be updated at 1KHz (but the update data can be generated from an extrapolation of a simulation on the order of 100Hz or even less in models where the stiffness contrasts between different model components/parts is not very large). Even at the lower end of the update range however, this places significant computational demands for models of even moderate complexity. This is compounded by two important desiderata that are essential for engineering CAD applications: large deformations (inducing nonlinear kinematic relations between strains and deformations) must be modeled, and topological change of the models must be allowed while still interacting with the models at haptic rates. Traditional continuum discretizations and standard numerical solution startegies are not adequate to support haptic interaction with changing models. The models generated tend to be prohibitively large and the solvers do not support the dynamically changing interactions and modifications of the model that are typical in the exploratory conceptual stages of design. Alternate models and numerical solution formulations are needed.
Two complementary ingredients comprise the deformation engine of ECAD: a simplified mechanics-based model---a surrogate model--- instead of the usual hexahedral or tetrahedral finite element discretizations, and an incremental formulation that allows dynamic boundary conditions, changes in model topology, and evolving nonlinearities to be expressed as low-order updates to the coefficient matrices to enable solution updates at haptic speeds. We describe these ingredients and illustrate their application in this section.

3.1 Skeleton-based Deformation Models

A common approach for performing force/deformation calculations in real time is to rely on a simplified representation of the model. Various mechanics-based models have been proposed for animation applications and surface deformations, including spring lattices and continuum finite elements with various energy terms simplified. Simplifications, naturally, degrade the accuracy of the results and the objective of any such strategy is to minimize the impact on accuracy while maintaining the flexibility and computational effectiveness of the simplified model.
Our simplification relies on the skeleton (medial-axis / medial surface) of the model. The skeleton not only has an appealing geometric representation as the main “spine” of a shape but when used with frame elements or shell elements can represent reasonably accurately the strain energy of the deforming solid. The line skeleton has been used in model reductions for the purposes of articulation and free-form geometric (non-physical) deformation in [2,17]. Here, we use the line and surface skeletons as the basis for constructing the mechanics model that guides the physical simulation. The main idea was described in an earlier work [1] and we summarize it briefly here.
The interior skeleton of a shape, defined to be the closure of the centers of maximally inscribable balls in the shape, is first generated from a generalized voronoi diagram of a boundary representation of the shape. This is then trimmed to

remove skeleton segments/sheets that terminate at the boundary, rediscretized, and the boundary-skeleton correspondence is established. Sample results on simple models are shown in Figs 1-2. Both in 2D and 3D, the skeleton captures the important characteristics of the model such as basic shape, topology, and thickness characteristics. Every point on the skeleton is associated with a radius indicating the thickness of the object at that skeletal point. Every point on the boundary is associated with a skeleton point along a direction normal to the boundary. Hence the position of a boundary point can be written as $x = sk(x) + r\,n$, where x is the position of the boundary point and $sk(x)$ is the position of the corresponding skeletal point along the boundary normal direction n, and r is the radius of the shape at that location (Fig. 3).

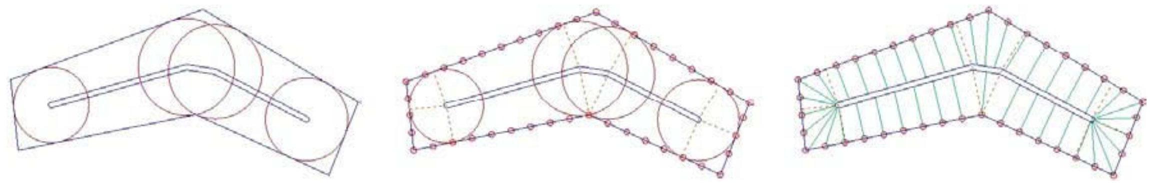

Fig. 1: A 2D model; its approximate skeleton; and skeleton-boundary association.

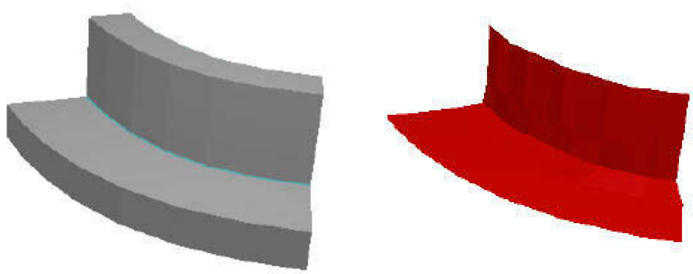

Fig. 2: 3D bracket and its skeleton.

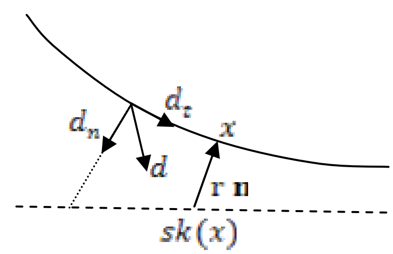

Fig. 3: Every boundary point is associated with a skeleton point. Every boundary displacement (d) may be decomposed into normal and tangential components.

To construct the surrogate model, every skeleton line segment (2D) or cell (3D) is then used to define a corresponding beam/shell finite element for mechanical simulation. The radius function, interpolated inside the elements from the values at the vertices, is used to set the axial and bending parameters of the element. When the skeleton deforms, the boundary of the object is moved along with it through the boundary-skeleton correspondence. The deformation of the skeleton is governed by the minimization of potential energy of the shape:

$$\pi(u) = \frac{1}{2}\int_{\Omega} \varepsilon : A : \varepsilon d\Omega + \frac{1}{2}\int_{\Omega} \kappa : B : \kappa d\Omega - \int_{\Omega} w u d\Omega$$

where u is the set of displacements and rotations at the skeletal nodes. ε and κ are the axial and bending strain tensors. In 2D they are scalar elongation and curvature changes along the line skeletons. In 3D they are the in-surface elongation and shearing strains, and the out-of-surface bending and twisting strains respectively. A and B are axial and bending constitutive properties which among other things depend on the radius function, and w represents the background forces acting on the model. The minimization problem is solved by the finite element discretization defined by the skeleton elements. Except for spatially varying properties, the finite elements we use are standard formulations. Formally, this surrogate model has the correct strain energy mechanical characteristics in the limit as the radius function approaches zero (thin objects). As the radius increases it is possible to add additional terms to the strain energy to account for shear deformations as well as compression through the thickness to generate more accurate models.

3.2 Modeling General Interaction

The basic haptic user interaction with the model consists of grabbing a point on the model and moving it in an arbitrary direction by an arbitrary amount, and directly feeling the force necessary to produce the deformation. Variants of this interaction with different tools to manipulate the shape are possible. The interaction is modeled as a dynamically changing boundary condition that sets the displacement of the grabbed point. The grabbed point may be on the skeleton (when the user wishes to manipulate the skeleton directly) but more often is a boundary point whose displacement has to be mapped to the displacement of a point on the skeleton. We again use the boundary-skeleton correspondence to find and set the displacements of the associated skeleton point. The user imposed boundary displacement is decomposed along the normal and tangential directions at the boundary (Fig. 3). The normal component sets the displacements of the skeletal point and the tangential displacement sets its rotations. Mathematically these displacement/rotation conditions on an arbitrary skeletal point are expressed as algebraic constraints containing linear combinations of displacements/rotations of the nodes of the element that the skeletal point lies in. Canonically they may be written as $Cu = g$, where C is a very sparse coefficient matrix, consisting of a few rows (one per displacement/rotation component constrained). The minimization problem described above is then solved subject to these constraints. In the linear case, this results in the following system of equations:

$$\begin{bmatrix} K & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix}$$

where K, the global stiffness matrix, is the Hessian of the potential energy expression and u is the set of displacements and rotations of the skeleton nodes from which the boundary deformation can be recovered. v is the set of Lagrange multipliers of the imposed constraints and corresponds to the set of forces required to impose the displacement of the grabbed point. They are rendered by the haptic device(s). This formulation has two key advantages. The first is that it preserves the structure K and hence can take advantage of any preprocessing done on it (such as a factorization). It also allows a partial decoupling between the solution for v (the haptic forces) and u (the displacements) and hence allows the haptic update to occur at a faster rate than the graphical update. This is done by rewriting the system above as:

$$CK^{-1}C^T v = CK^{-1}f - g$$

$$Ku = f - C^T v$$

The first equation can be solved multiple times (haptic update) before the second one (graphics update) is solved. The triple product results in a very small set of equations with a very sparse constraint matrix C. A preprocessing of K (factorization or inverse) is obviously very advantageous since g and C are the only quantities changing during interactions that do not modify the model. g is a simple transformation of the movement of the haptic device to the displacement and rotations of the affected nodes. As the region of contact between the haptic end and the solid model changes, C is updated and the first equation above is solved allowing, for example, the user to roll the stylus of the haptic device on the model boundary and feel continuously its spatially varying compliance.

When the model exhibits large deformations, nonlinearities in the kinematics must be taken into account and the stiffness matrix K must be updated at runtime. A complete refactorization of the stiffness matrix at haptically-acceptable rates is only possible if the model consists of relatively few, say a few hundred, elements. For most CAD parts of interest a reasonably realistic prediction of deformation behavior requires at least an order of magnitude more elements, and therefore the regeneration of the factorization must be performed using different strategies.

An incremental update of the factorization can be obtained by observing that the spatial extent of the nonlinear region changes incrementally. In most interactions, one element or very few elements per solution cycle exhibit large enough strain change that requires a change in their stiffness matrices. Therefore at every cycle of the haptic force solver, one or a few element matrices must be updated. The update to the global stiffness matrix is then expressed as:

$$K_{new} = K + \sum a^t \Delta k a$$

where the sum ranges over the elements affected, Δk is the change in the affected element stiffness matrix and a is the compatibility matrix that locates the element stiffness matrix in the global stiffness matrix. The form of this update allows the use of the Woodbury formula [5] to update the factorization of K. The update consists strictly of products of sparse matrices and very sparse vectors and therefore can be performed in O(n m), where n is the total number of degrees of freedom in the model and m is the number of elements that need to be updated (typically one or very few). Similarly when the thickness of various elements change or when new elements are added, a similar update computation can be performed on the stiffness matrix. This strategy has allowed us to develop a system responsive at haptic rates for models with a few thousand dynamically modifiable elements on standard desktop machines. We describe the system and its interface next.

4. ECAD PROTOTYPE

Our main design criterion for ECAD is that it has to be a strict superset of existing CAD systems. We believe that only modular enhancements to the standard interfaces that are in use by commercial CAD systems are likely to be viable for broad adoption. There is far too much invested in current editing techniques and interface metaphors to be able to allow for a radical reengineering of the CAD design experience in the short term. As a result, the ECAD system has the usual 2D menus, toolbars, design tree, etc. and augments them with a 3D workspace. Both from a UI point of view and interaction point of view the system mixes 2D and 3D interface elements.

When a 3D stereo display and a haptic device and auxiliary tracking and 6 DOF pointing devices are plugged in, the user can experience the richer interactive environment of ECAD. As we describe below, this environment makes a number of editing tasks easier and more intuitive to accomplish with richer 3D design tools, and provides tangible feedback about the mechanical behavior of the model using the real-time deformation engine described in the previous section. Without these devices, the 3D workspace portion of the ECAD environment regresses to the usual mouse driven workspace.

4.1 User Interface Components

Fig. 4 shows the ECAD system in use. Its components include: (A) an ultrasonic headtracker signal emitter; (B) a PHANToM haptic I/O device; (C) Shutter glasses with ultrasonic headtracker signal receiver; and (D) a Spaceball 6DOF mouse. The Phantom is the primary input device in the ECAD system. It serves as a 3D mouse through which the user performs input, selection, and manipulation tasks. When the Phantom pointer is within the 3D workspace portion of the UI, it acts as a 6 DOF input device. When the pointer exits this area, only two degrees of freedom are used and it acts like a standard mouse, allowing the user to perform menu selection, toolbar interactions, and all of the other actions typical of the familiar WIMP (Windows Icons Menus Pointer) metaphor. The Spaceball is used to manipulate the 3D view of the model. The Spaceball is always active and two-handed interaction is encouraged. The Spaceball can be used to adjust the view of the part while the Phantom is used to simultaneously manipulate geometric elements of the model. This is particularly useful for performing selection and control point manipulation tasks.

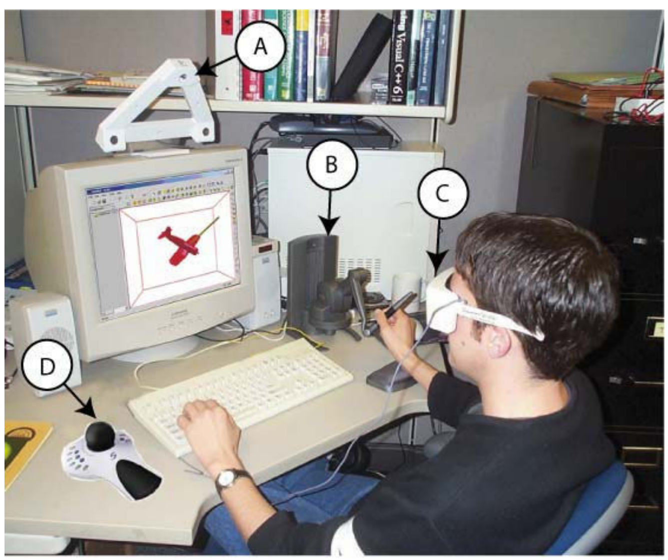

Fig. 4: 3D UI hardware incorporating an ultrasonic tracker (A), a PHANToM stylus (B), shutter glasses (C), and a Spaceball (D).

Like the Spaceball, the head tracking system is used for view manipulation. The use of the head tracker allows the application to update the user's viewpoint based on head position and orientation in a very natural way. The system consists of two components: A group of three ultrasonic emitters mounted on a triangular frame (Fig 4, A) and three

receivers mounted on the shutter glasses (Fig 4, C). ECAD uses this information to define a viewing coordinate system from the user's position and orientation in front of the monitor and "stabilize" the 3D view with respect to the user's viewpoint. Without head tracking the virtual image would always be rendered from a fixed viewpoint, which might not correspond to the user's actual position, causing the image to seemingly "follow" the user around rather than appear solid and stable. Head tracking also improves the user's 3D perception by taking advantage of the parallax depth cues that arise from moving around in front of the model. In some situations Ware et al. [15] found that head tracking is sometimes more important to spatial awareness than stereoscopy.
The output portion of the ECAD interface consists of two main components: the 3D display and the Phantom haptic force-rendering device. The coupling of a 3D input device with the 3D display is an essential feature of the ECAD interface. Many programs available today incorporate 3D display capabilities, and while this does improve the user's awareness of the 3D model greatly and makes some operations more intuitive, the full implications of a 3D view are not realized until it is coupled with a 3D input device. Many operations such as selection and manipulation are greatly enhanced with a 3D input device and 3D view. ECAD provides the user with a three-dimensional display via a monitor and shutter glasses. Using well-known techniques, the ECAD graphical renderer generates separate left and right eye views, which are rapidly alternated on the screen. The shutter glasses alternately black out the left, then the right eyepiece in synch with the monitor update. The end result is that each eye sees the appropriate image to produce depth perception.

Fig. 5 shows a screenshot of the ECAD interface with the primary UI components labeled:

- (1) The 3D Work Volume. The dominant feature of the ECAD interface is the 3D working volume. Within this space, the 3D model is displayed and the user interacts with it via the a 6DOF cursor controlled by the PHANToM. The extents of the working volume are indicated by a virtual wireframe box which sits inside the monitor with its front face coplanar with the screen. Objects are not clipped against this volume and may protrude beyond it in any direction, but this volume denotes the comfortable working extremes of the Phantom's range of motion and generally the user will work inside of this area. The view is presented in head tracked 3D stereo.
- (2) Cursor. The cursor is the primary interface for interacting with the model and the UI. When in the 3D work volume, the cursor is displayed as a 3D pencil, as shown in the figure. Whenever the cursor leaves the 3D work volume, the standard Windows 2D cursor is displayed.
- (3) The Workplane, Feature, Component Tree Pane. The pane on the left hand side of the window contains one of three tree views selectable by the user which display workplanes & sketches, model features, or assembly components. These trees can be used for selecting model entities, organizing the data with user-friendly names, reordering feature generation, and other operations similar to what is offered by most commercially available feature-based modelers.
- (4) Menus, toolbars, status bar, etc. The rest of the ECAD interface consists of standard windows components, which are used in the usual manner. One subtle, yet important feature to notice is how the 2D and 3D interface components integrate seamlessly with one another. The use of shutter glasses for 3D requires that both a left and a right eye image be generated and displayed for the user. 3D graphics APIs provide straightforward functions for doing this for the 3D model, however there is no obvious way to do this for the 2D interface components that are managed by the operating system. However if we make the decision that all 2D interface components will reside in the zero-parallax plane (i.e. the plane of the monitor screen) then the left and right eye images become identical and we avoid the need to generate two separate images.

4.2 Haptic Enhancement to Modeling Tasks

The ECAD system supports a number of two-handed, 3D input and haptic interactions to facilitate common modeling operations and tasks. Object and view manipulation, for instance, are greatly enhanced when the 6DOF spaceball, manipulated by the non-dominant hand, is used to control viewpoint position and orientation, or when the stylus of the PHANToM is twisted to zoom with one simple interaction. Operations such as sketching on object facets, selecting occluded features, mating faces, etc. are also made much easier when the haptic channel is used. Depth perception makes freeform surface and control point selection and manipulation far more accurate with fewer viewpoint changes. When experienced *collectively,* these simple enhancements to the bulk of user interaction movements significantly increase the overall effectiveness of the environment. We briefly highlight some of the haptic enhancements next.

Touchable Objects. This is probably the most obvious application of force feedback technology to solid modeling. We can readily simulate touching and feeling the compliance of the object by resisting the user's movement on or below the surface of the model as computed by the deformation engine. In the limit, rigid objects provide constant force feedback with no deformation when touched.

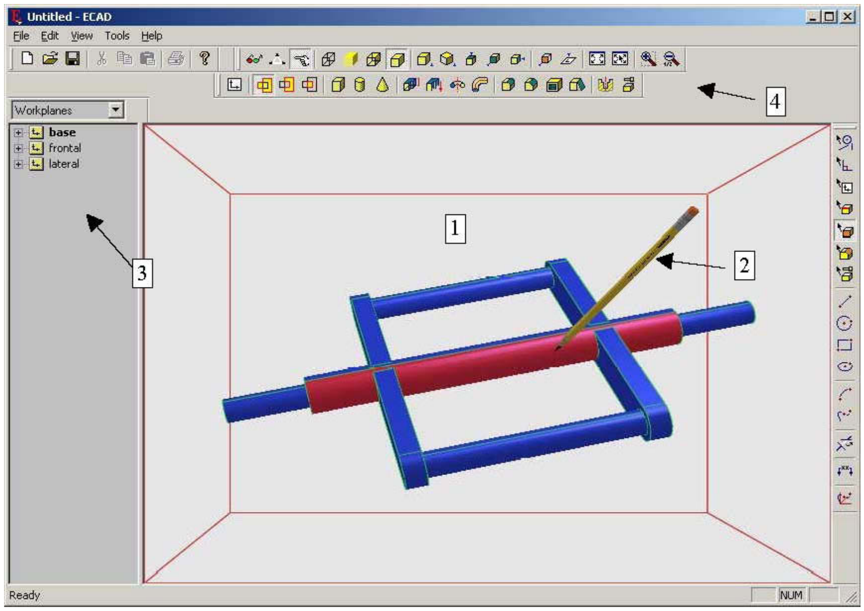

Fig. 5: ECAD Interface incorporating 3D elements and 2D elements (residing in the zero-parallax plane).

Haptic Transparency. Haptic rendering of the model as a deformable or rigid solid is not always desirable. The solid object can prevent the user accessing all portions of the model, or simply get in the way. However, it is still useful to have an indication of whether the stylus lies inside or outside the object. One useful interaction mode we have developed is "haptic transparency." When the object is haptically rendered in a "transparent" state, an inertia effect is applied to the stylus whenever it is inside of a solid body. The result is a feeling of viscosity or thickness when inside of the model. This feature is implemented by applying a force to the stylus that opposes acceleration and dampens velocity when the tip of the stylus is inside the model. This is particularly useful for selecting interior features (Fig. 6).

Restricting DOF to match the task at hand. Any time the user is supplied with more degrees of freedom than the task requires, the complexity and opportunity for error increases. One important use of haptics in the ECAD environment is to restrict the degrees of freedom to suit the task at hand. For example, creating a 2D sketch (for extrusion, for example) is a 2D task and it is made much more difficult with a full six degrees of freedom (imagine trying to write on an unsupported sheet of paper in mid-air). Proper use of haptic feedback allows ECAD to simulate solid work planes upon which the user can sketch comfortably and intuitively.

Snapping and Grooved Sketch Planes. Most modern sketching interfaces include some sort of user-configurable snapping functionality that can be used to cause the cursor to automatically snap to significant locations within the sketch. These significant points are referred to as snap points. For example, if the user is defining the endpoint of a line, the cursor may snap to the endpoints or midpoints of existing lines, or perhaps to parallel and perpendicular configurations. This seems like a natural candidate for haptic enhancement. However, the obvious implementation of haptic snapping whereby a force is applied to the stylus to draw it towards snap points is not really very helpful and more often than the user finds that they are fighting against this force and their productivity is decreased rather than enhanced. A better solution to the snapping problem for 2D sketching on arbitrary planes is to define *grooved sketch planes* by making use of the extra degrees of freedom provided by the stylus to modify the snapping behavior on the fly. Snapping in ECAD uses the pressure of the stylus on the sketch plane to modulate the snap force. The harder the user presses on the sketch plane, the stronger the attraction to snap points. The result is that the user feels as though the sketch plane has V-shaped grooves around snap points. Using the grooved sketch planes approach allows users to determine the magnitude of the snapping action to suit their needs on the fly (Fig. 6).

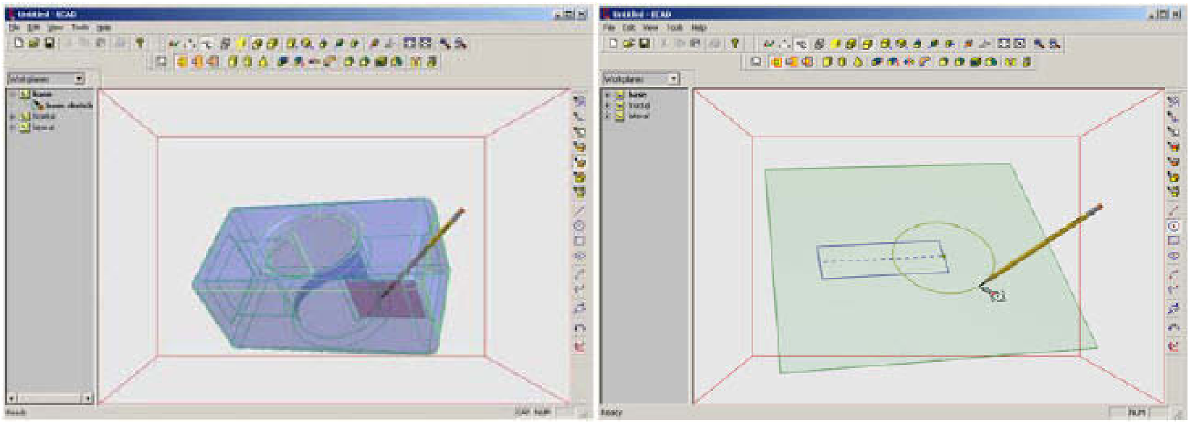

Fig. 6: Examples of model editing enhancements made feasible by haptic feedback: haptic transparency to aid in interior feature selection (left), and haptic sketch plane and grooves to provide a natural snapping behavior (right).

4.3 Examples

Figs 7-12 show a number of examples of interactions with compliant models. Figs 7 and 8 are 2D models of a gripper and a 2D spring respectively. The automatically generated skeletal models that define the mechanical interaction models are shown. The user pushes on the models, that undergo large deformations, to assess their relative stiffness. Figs 9 and 10 show similar interaction with 3D models. The 3D geometric models are designed in the ECAD environment and become automatically compliant using their skeletons. The haptic device can then be used to feel the overall stiffness of the models as they undergo nonlinear deformations. Fig. 11 shows snapshots of a design session where the user is modifying the shape of a crimper in response to the tangible feedback they get on model flexibility. The shape is stiffened in one region and its weight reduced in another to obtain the design shown in Fig 12. Essentially, the user is solving the implied shape optimization problem by *physically* interacting with the virtual crimper.

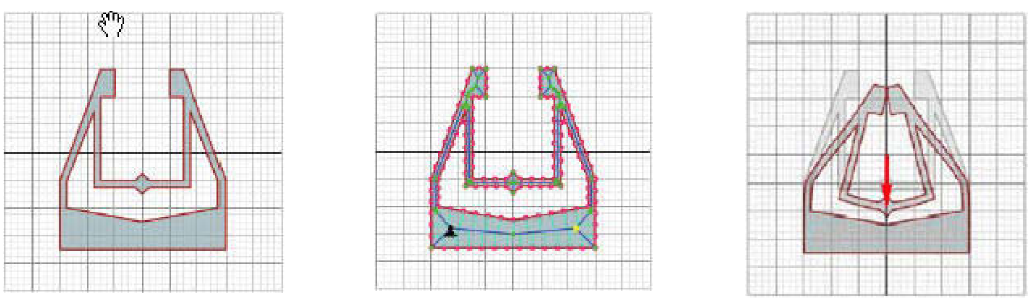

Fig. 7: Haptic interaction with a compliant gripper.

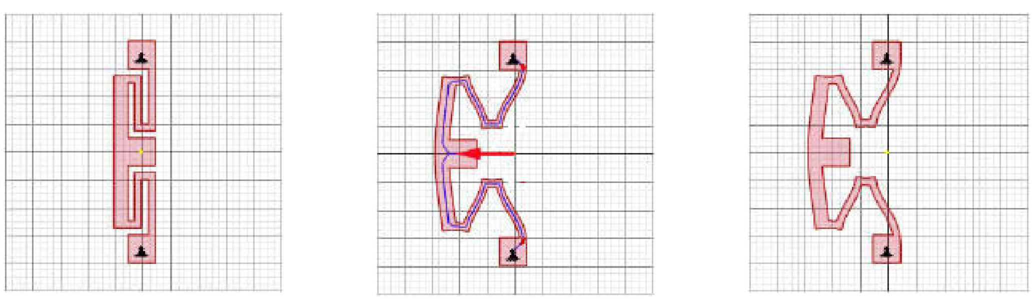

Fig. 8: Haptic interaction with a 2D spring.

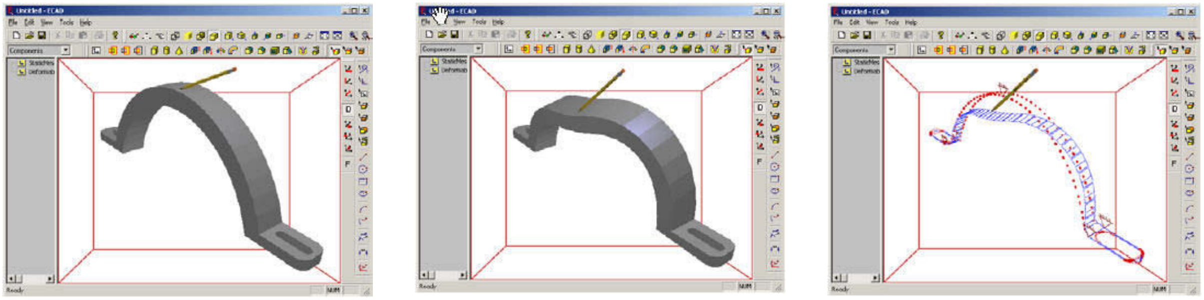

Fig. 9: Haptic interaction with a 3D bracket.

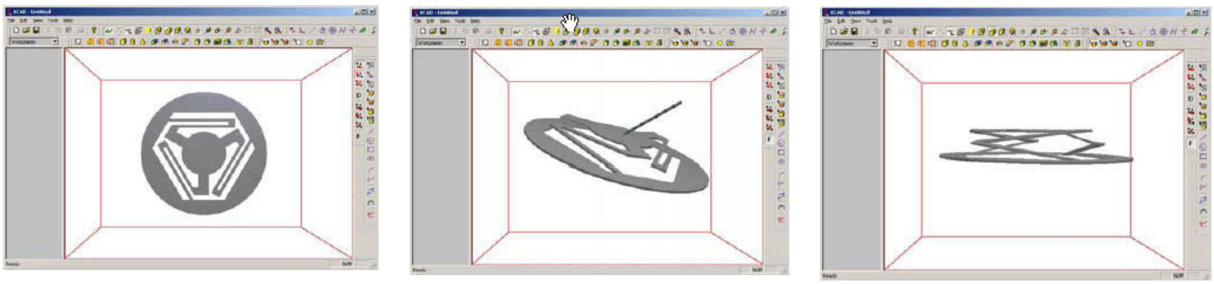

Fig. 10: Haptic interaction with an ortho-planar linear-motion spring.

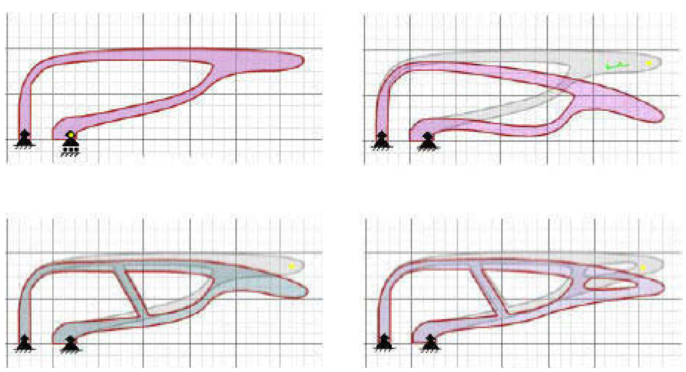

Fig. 11: Real-time shape modifications to improve stiffness to weight ratio.

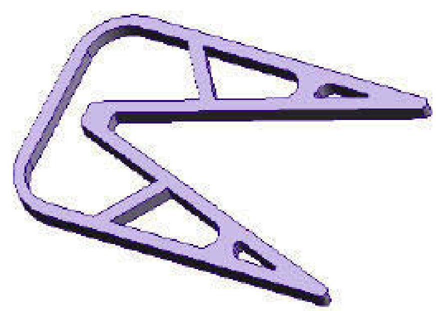

Fig. 12: Final crimper design.

5. CONCLUSIONS

We described a proof-of-concept CAD environment that incorporates next-generation interface devices in a way that greatly facilitates creating and interacting with 3D solid models. The ECAD system brings together a 3D head tracked display and 6 DOF input devices to form a screen-based VR modeling environment that enables more accurate and intuitive interactions. Haptic interface devices add another dimension of interaction to the modeling environment that can be exploited in nearly every aspect of modeling from sketching to simulation. Coupled with a real-time deformation engine, haptic feedback also allows the user to tactilely assess the compliance characteristics of 3D models.

Compared to immersive systems with their expensive and awkward hardware the screen-based VR system has the advantage of allowing a user to sit comfortably at a conventional workstation and only requires the addition of a few relatively inexpensive and comfortable devices. By integrating these new interface tools into a familiar Windows-like environment we retain and extend the functionality of existing interfaces rather than require the user to learn an entirely new interface metaphor. We believe that interfaces similar to the ECAD environment have significant potential within the CAD industry, particularly in the area of conceptual design and virtual prototyping.

6. REFERENCES

[1] Blanding, R.; Turkiyyah, G.: Flexible bones for haptic prototyping of deformable objects, Virtual Reality, 7(2), 2004, 103-111.

[2] Bloomenthal, J.: Medial-based vertex deformation, ACM Symposium on Computer Animation, 2002, 147-151.

[3] Botsch, M.; Kobbelt, L.: An intuitive framework for real-time freeform modeling, ACM Trans. on Graphics, 23 (3), 2004, 630-634.

[4] Butterworth, J. et al: 3DM: A 3D modeler using a head mounted display, Symposium on Interactive 3D Graphics, 1992, 135-138.

[5] Golub, G.; Van Loan, C.: Matrix Computations, Johns Hopkins, 1996.

[6] Gribnau, M.; Pasman, G. TIME: Three Dimensional Input, Modification, and Evaluation, ACM Conference on Human Factors in Computing Systems, 1996, 400-401.

[7] ayaram, S.; Connacher, H.; Lyons, K.: Virtual assembly using virtual reality techniques, Computer-Aided Design, 29 (8), 1997, 575-584.

[8] Liu, X.; Dodds, G.; McCartney, J.; Hinds, B.: Virtual DesignWorks---Designing 3D CAD Models via Haptic Interaction, Computer-Aided Design, 36, 2004, 1129-1140.

[9] McDonnell, K.; Hong, Q.: Dynamic sculpting and animation of free-form subdivision solids, Proceedings IEEE Computer Animation, 2000, 126.

[10] Peng, X. et al: Free Form Modeling using sweep differential equation with haptic interface, Virtual and Physical Prototyping, 1(3), 2006, 183-196.

[11] Piegl, L.: Ten challenges in computer-aided design, Computer-Aided-Design, 37, 2005, 461-470.

[12] Sachs, E.; Roberts, A.; Stoops, D.: 3-Draw: A Tool for designing 3D shapes. IEEE Computer Graphics and Applications, 11(6), 1991, 18-26.

[13] Salsedo, F. et al.: Architectural Design of the Haptex System, HAPTEX 05 Workshop on Haptic and Tactile Perception of Deformable Objects, 2005, 1-17.

[14] Sederberg, T.; Parry, S.: Free-form deformation of solid geometric models, ACM Trans. on Graphics, 20, 1986, 151-160.

[15] Ware, C. et al: Fish Tank Virtual Reality, ACM Conference on Human Factors in Computing, 1993, 37-42.

[16] Ye, X.; Campbell, X.: Supporting Conceptual Design with Multiple VR Interfaces, Virtual and Physical Prototyping, 1 (3). 2006, 171-181.

[17] Yoshizawa, S.; Belyaev, A.; Seidel, H.-P.: Free-form skeleton-driven mesh deformations, ACM Symposium on Solid Modeling and Applications, 2003, 247-253.

[18] Zhou, K. et al: Large deformation using the volumetric graph Laplacian, ACM Trans. on Graphics, 24 (3), 2005, 496-503.

[19] Zhu, W.; Lee, Y.: 5-axis pencil-cut planning & virtual prototyping with 5 DOF haptic interface, Computer-Aided Design, 36, 2004, 1295-1307.