Feeling and Seeing: Issues in Force Display

Margaret Minsky*†, Ming Ouh-young*, Oliver Steele*1,
Frederick P. Brooks, Jr. *, Max Behensky2

*Department of Computer Science,
University of North Carolina at Chapel Hill
Chapel Hill, NC 27599

†MIT Media Laboratory
20 Ames St.
Cambridge, MA 02141

Introduction

This paper is about using the sense of touch, the haptic system, as part of our everyday interface with computationally created worlds.

In Part I, we discuss a particular system, called *Sandpaper,* designed for experimenting with feeling texture. In part II, we discuss how control analysis helps us understand the behavior of the types of hardware and software we use to implement force display.

We will not comprehensively review the literature of force display devices and their applications; descriptions can be found in e.g. [Sutherland65], [Noll72], [Batter72], [Atkinson77], [Brooks88], [Ouh-young88], [Smith88].

Part I

Force display technology works by using mechanical actuators to apply forces to the user. By simulating the physics of the user's virtual world, we compute these forces in real-time, then send them to the actuators so that the user feels them. The force display technology we use in the *Sandpaper* system is a motor-driven two-degree of freedom joystick (built by Max Behensky and Doug Milliken). The joystick position is reported to the software, which computes the appropriate forces for the joystick's motors.

Why create texture?

Force display is especially useful for communicating surface texture and bulk properties of objects and environments as well as dynamics of objects. In this way force display technologies augment the strengths of computer-generated graphics and sound in creating convincingly realistic environments (figures 1, 4).

In the *Sandpaper* system, we use a novel technique to allow the user to feel textures. We create very small virtual springs which pull the user's hand toward low regions and away from high regions of a texture's depth map. We synthesize finely spaced grooved surfaces and also use depth data from Perlin's noise textures [Perlin85] and fractals supplied by Pentland [Pentland84]. We also create feel-able physics such as variable viscosity soups, springs, and yo-yos (Color plate 1).

We believe it is particularly important to allow the user to make exploratory motions as if they were touching real objects and materials [see Lederman87]. This informs our empirical studies and our design of future force display devices.

Surfaces as perceived by the human haptic system derive from a complex combination of shape and material properties. Texture is one of the most important such properties. What is salient about a surface may also involve other percepts, such as softness, apparent temperature, and so forth [Katz25]. We believe that we can make computer interface systems which can synthesize all of these; in order to do that we need to understand both the perceptual and computational issues.

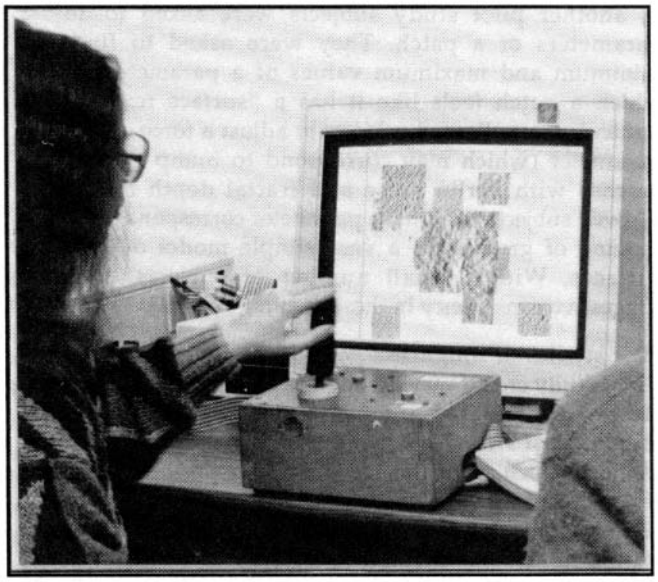


Figure 1 Photo of "Sandpaper" System In Use

Empirical Studies of Roughness Perception

Our application is a computer-supported software and hardware system to do research on texture perception.

---

[1] Present address: Apple Computer, Inc., Cupertino, CA
[2] Address: Atari Games, Milpitas, CA

The system was designed as a laboratory for conducting experiments on touch and haptic perception. An experiment we envisioned is the following: the user is presented with several pieces of sandpaper and is asked to arrange them in order of roughness, exploring them by finger touch, and sliding them on a table in front of him. This experiment takes place in a virtual environment using a force feedback interface; the textures are created computationally with characteristics controlled by software. (Our experiments are adapted and simplified from psychophysical studies by Lederman and others; see, for example, [Lederman72].)

We created patches of texture intended for testing roughness perception. There were two major goals: first, to provide a flexible testbed for testing human perception of roughness and for creating the stimuli for such experiments; and second, to understand differences between the perception of computationally created textures and the perception of real textures in the interest of creating textures which sustain the illusion of objects in virtual realities.

Evidence that texture simulation works

In a pilot study, subjects believed that they were feeling patches of textured material.

In an anecdotal experiment, subjects were presented with a variety of textured patches (visually masked) in the *Sandpaper* environment. They were asked to arrange the patches in order of roughness. Subjects ordered the patches with a moderate degree of consistency. We interpret this to mean that subjects are able to judge roughness of these patches, and can discriminate degrees of roughness. Since we use a variety of surface models, we do not know which aspects of the simulated surfaces contribute to the perceived roughness.

In another pilot study subjects were asked to adjust parameters of a patch. They were asked to find the minimum and maximum values of a parameter within which a patch feels like it has a "surface texture". In particular, we allowed subjects to adjust a force amplitude parameter (which may correspond to bump depth), for patches with Perlin noise and fractal depth maps. We allowed subjects to adjust a parameter corresponding to the spacing of grooves in a very simple model of grooved surfaces. With a small number of subjects, we saw suggestive consistency in these results.

Pilot studies of roughness perception suggest that we can successfully create varying degrees of perceived roughness. However, the spatial parameters that determine perceived roughness of virtual grooved surfaces may not be the same as those which are correlated with perceived roughness for real grooved surfaces as observed by Lederman and Taylor [Lederman72]. Preliminary data suggests that roughness of simulated surfaces may be closely correlated with spatial frequency, although there are several other likely hypotheses.

How to Create Simulated Textures

First, we oversimplify by saying that texture is made of little bumps. The little bumps are surface features in the range from a few microns up to millimeters.

Previously, we had created the illusion of bumps and valleys by the following trick: As the user moves the joystick in a direction which is "up" a bump, his motion is opposed by a spring force proportional to the height of the bump. This gives the sense that it is very difficult to move to the top of the bump (springs resist being stretched), and easy to fall off the bump back into a lower region of the simulated surface (springs like to revert to a short length).

We had made fairly large scale bumps whose apparent heights were determined by the stiffness of the surrounding springs. This technique has been used by several groups working with force display technology to create artificial detents (spring force toward the bottom of the detent).

We extended this technique to fine grained surfaces by computing spring forces based on a local gradient: As the user moves the joystick on the virtual surface, the change in height in the direction of motion is noted. We create virtual springs opposing the motion "up" the sides of each tiny bump. Thus the spring forces applied to the hand are computed from local gradients of the height of the surface (figures 2,3).

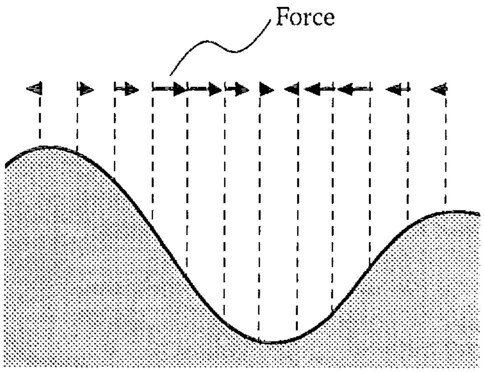


Figure 2 Gradient Technique: enlarged cross section of surface depth map

The texture forces are computed in real-time from a texture depth map. In some cases the depth map is stored and in some cases it is computed on the fly from a procedural representation of the texture.

Our system associates a screen picture with each patch. Usually we associate a shaded rendering of the appropriate texture with what the user feels as he manipulates the patch. Sometimes, for purposes of our experiments, we "paint over" the patch to give little visual feedback; sometimes we even display conflicting visual information.

$$F_x = A\frac{\Delta h}{\Delta x}$$

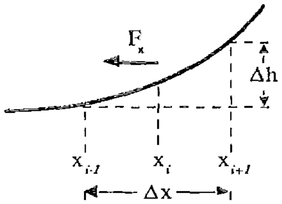


Figure 3 Gradient technique: detail of local spring force computation for x direction; y direction is similar

Force Display Requires Real-Time Physics and Animation

The approach outlined above for simulating texture is a combination of lookup of material properties and real-time computation of system dynamics. In the texture computations described above, we compute forces locally.

Textbook objects

This approach implies a physical interpretation of the algorithm for generating forces. Any physical system can be created from a combination of ideal "textbook" objects: springs, dampers, and masses:

Spring	**Force = k*position.**
Damper	**Force = b*velocity**
Mass	**Force = m*acceleration**

For textures, we currently compute forces based on local position information, which we then interpret as virtual springs. Sometimes we compute other dynamics as well in order to simulate material properties other than local slope. For example, we sometimes apply viscous damping forces to stabilize our simulation (see Part II).

In fact, in our system we model a variety of non-textural physical systems as well. We needed to perfect our models of pure springs, dampers, and masses in order to create textures, and we also have patches of materials such as molasses and ice, and patches containing bricks to push, lassos to twirl, and independently moving objects which try to drag the user's hand about.

In general, to model *any* dynamic system, we model a combination of the user and the force display device itself. We must sense the position, velocity, and acceleration of the joystick, and then use geometry and equations of motion to compute the appropriate output forces.

We can compute modest physical setups and animated displays in real time. Sometimes it is easier or necessary to precompute forces, or some other aspect of the environment that will be queried to produce the forces. For example, a square-wave grooved surface should be easy to produce purely procedurally, but its very steep walls cause local instabilities in the physics simulation (see Part II). We have to filter the depth map of the grooved surface in advance. In general, this kind of filtering is too slow to do in real time.

Physical modeling in a sampled, digital world requires attention to signal processing techniques. We learned several techniques which allowed us to increase fidelity to real-world physics and stability. The spatial filtering mentioned above is an example. We also temporally smooth our acceleration and velocity data. This can smooth the feel of the simulation, at the expense of increased apparent viscosity, mass, and lag. Thus some techniques required us to make tradeoffs to achieve reasonable results.

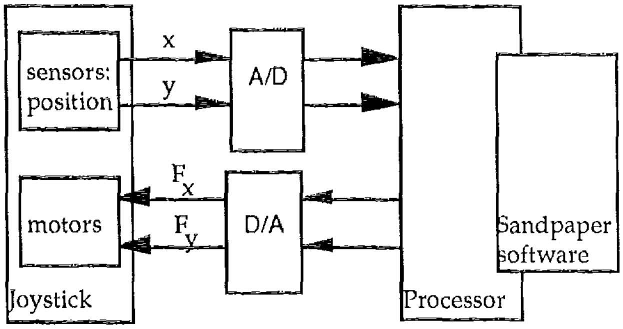


Figure 4 Block Diagram of Hardware System

Patches of Feel-able Material are Objects

Since our primary aim was to create moveable patches of textured surfaces, we decided to represent each patch visually as a rectangle on the screen that could be manipulated by the user.

Internally, a patch is an object with state. Each patch is associated with either a procedure that runs each clock tick to produce forces, or with an array of precomputed force values to be looked up on the basis of joystick position. The patch is associated with its visual representation, a pattern or a bitmap.

Each patch has several parameters. For example, a patch representing a finely grooved surface has parameters for the height and spacing of the grooves. A patch may also have other parameters: for example, the viscosity of the virtual medium in which the user's hand is moving, or the apparent mass of the joystick. The *Sandpaper* system allows any of these parameters to be controlled by screen-based sliders which the user can adjust.

This approach to the design of the system was vital for the kind of empirical work now required. For example, in creating a simulation of a mass (think of wielding a brick on the end of the joystick) there were several parameters whose values were critical (mass, viscosity, spring constant, and temporal filter coefficients). By tuning these parameters in an interactive loop, and *feeling* the response of the system, we were able to create stable masses. However, our experiences with empirically varying ill understood parameters prepared us to appreciate the analyses offered in Part II!

The ability to create patches in the same family, but with different parameter settings, and to vary the parameters of a patch in real time are important not only to the implementor of virtual materials, but also to the experimental psychologist who is a user of our system. For example, an experiment can be designed in which the subject is asked to vary a texture until it matches a reference.

How Much Must the Environment Suggest Physical Reality?

Interface metaphor and graphics must enhance the touch illusion of virtual patches of textured material that can be directly manipulated by the user. We used direct manipulation conventions from the Macintosh interface as much as possible. A challenge was to keep the visual contents of the patches constant as they are dragged; this quality of animation is vital for a sense of physical reality.

We tried various experiments to increase the sense of physical reality. What should happen when the user wants to move a piece of sandpaper? Should he be able to pick it up? We simulated this by having it become the topmost layer. Should it slide? We simulated this by maintaining a two-and-a-half-d ordering of the patches according to the order in which they were originally placed on the table. Each of these strategies gives the user a different impression of the physical environment in which he is manipulating the material. These are visual strategies. The next step is to model the physical interaction forces between the patches so that heavier materials are harder to drag than light ones, and rough materials are harder to drag past each other than smooth ones.

The haptic sense as it is used to explore an environment, is a combination of cutaneous and kinesthetic senses with intentional exploratory motions. We believe that users' freedom to perform the appropriate exploratory motions strongly affects their perception of a simulated physical environment.

Lederman and Klatzky [Lederman87] assert that stereotypical hand motions are associated with exploring objects for certain features. In particular, "lateral motion" is associated with texture. In pilot studies within the *Sandpaper* environment, we observed that subjects used this stereotypical motion when asked questions about texture properties of patches. Subjects complained in a variety of ways when patches were too small to allow this motion while staying within the patch; making the patches bigger removed these vague discomforts.

The joystick interface raises the question of whether perception in the *Sandpaper* environment is more like perception of objects directly with the hand or perception through a handheld tool. Although the joystick ostensibly resembles a handheld tool, we have had some success in bringing the perceived location of our textured surfaces very close to the hand, by building an apparatus that changes the physical appearance of the joystick and the way the hand can grip it.We cover the joystick with a black box, and mount a ping-pong ball on its end. The ball appears to be sliding a flat black surface. About one half of our subjects perceive the textured patches with this arrangement to be directly beneath the hand.

Conclusion: Moving to higher level descriptions

We have emphasized description of texture in terms of mechanical impedances and low level physical models. In fact, geometric and physical models are our way now to implement feel-able physical and shape properties of objects.

It is important to move to descriptions at higher levels including those informed by perceptual dimensions, detailed knowledge of which we hope to gain through further perception experiments. For example, we would like to describe a surface in terms of degrees of roughness, softness, and stickiness rather than in terms of density and placement of tiny bumps.

Our planned perception experiments should uncover mappings between these percepts and the physical parameters of our simulations.

We can then make higher level descriptive building blocks to create full three dimensional virtual worlds containing passive and active objects and ambient media.

Future

Further quantitative studies on texture perception must be performed using simulated surfaces. In particular, we are developing better models of grooved surfaces to be used in studying roughness perception.

We will report on studies of the roles of texture and *other* properties of simulated objects perceived by the haptic system in virtual worlds. We emphasize observations of the motions made by users, in order to understand how exploratory activities used in the real world may be used in virtual worlds. It is as important to integrate our system with sound cues as with visual cues; that is part of our work in progress.

We are beginning work with a three degree of freedom joystick (x,y,and z) [Smith88, Russo90], and will compare our two-degree-of-freedom results with those obtained in true 3D. We expect to map our textures onto the surfaces of three dimensional objects, and also to create soft surfaces and "volume texture".

We will report progress and recommendations in using force display in particular virtual worlds; for example, teleoperation, surgical simulation, and sculpture.

Part II: Control Issues in Force Display

To analyze the methods for creating an illusion of feel in Part I, we investigated a variety of control issues in force display:

1. Analyses to address the question: What is the required system updating rate for system A doing simulation B? Analyses using control theory yield the stability conditions among sampling period, mass, stiffness, and viscosity in various simulations.
2. Measurements on a force-feedback joystick and an ARM (Argonne remote manipulator). The experimental data support our predictions from theory. We used the Sandpaper system to conduct several interesting experiments, and used the analyses in control theory to explain these strange phenomena (some of which are counter-intuitive).
3. Measurements on the human arm. We followed Hogan's approach and found that there is a significant difference in human arm impedance between radial motion (forward-backward) and tangential motion (side-to-side) when holding a joystick [Hogan89].

What destroys an illusion of feel?

In general, data conversion and computer speed limits the attainable sampling rate in force display. For real-time computer graphics, 15-30 frames/second performance proves enough for acceptable visual illusions. What is the minimum sampling rate for good perception of force through a human arm? This question relates to human response time, human arm dynamics, system performance, and what is being simulated.

One thing is certain, if the system is inherently unstable, the illusion of a real object is destroyed immediately. Another criterion common both to computer-generated images and computer-synthesized force fields is that the displayed object cannot jitter if it is supposed to be stationary in time. Noise (quantization noise, thermal noise in potentiometers, noise in transmission lines, and noise in electronic components) in input data causes jitter problems. Bad force-field simulation/system dynamics causes inherent instability, which is intimately related to the sampling period.

Impedance control theory

How to create an illusion that one is holding a real object? Let's be more specific. Given a computer controlled joystick, can we simulate the dynamics of a spring-mass system including its mechanical impedance (mass, stiffness, viscosity)? If we can simulate the mechanical impedance then one cannot tell whether an object is real (using tests of Newton's three laws of motion). In theory, we can do this simulation, if we can measure the position, velocity, and acceleration of the hand-controller precisely, and can calculate and deliver the outputs continuously and without lag.

The following analysis follows the impedance control theory introduced in [Hogan87]. Suppose the joystick has mass m, stiffness k, and viscosity b. Define position as x, velocity as v, acceleration as a, the force generated by the motor controlled by a computer as F_s, and force measured by a force sensor as F_{ext} (Fig. 5).

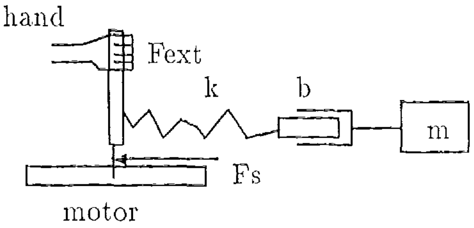


Figure 5: A joystick system

$$ma + bv + kx = F_s - F_{ext} \tag{1}$$

Suppose the target virtual spring-mass system has mass M, stiffness K, and viscosity B, then the force measured at the sensor is

$$-F_{ext} = Ma + Bv + K(x - x_0) \tag{2}$$

where x_0 is the rest position. From 1, the force required at motor is

$$F_s = ma + kx + bv + F_{ext} \tag{3}$$

from 2, let $1/M = W$

$$a = W[K(x_0 - x) - Bv - F_{ext}] \tag{4}$$

substituting 4 into 3

$$F_s = mW[K(x_0 - x) - Bv] + kx + bv + F_{ext}(1 - mW) \tag{5}$$

Equation 5 says that if the position x, velocity v, and the force from sensor F_{ext} can be measured, the system can simulate any object by controlling motor forces only.

A method for creating an illusion of feel

The goal is to simulate a spring with stiffness K, mass M = m, and no viscosity. Instead of doing the detailed simulation in the ideal case, we choose to use a very simple method: let the joystick synthesize the spring force based on position feedback only. The question becomes, what does the human arm really feel?

Let $F_s = K(x_0 - x)$ in our simulation. From Eq. 5, assuming joystick stiffness (without power) is zero.

$$\begin{aligned} -F_{ext} &= -[K(x_0 - x) - ma - bv] \\ &= ma + bv + K(x - x0) \end{aligned} \quad (6)$$

The true behavior of the system, and so the feel to the human arm as an external observer, is like holding a spring with stiffness K, mass m, and viscosity b. Although this is not exactly the target spring system (mass m, stiffness K,viscosity 0) in simulation, it is close to the target system if viscosity b is small.

With this simple approach, we successfully built dynamic models in Sandpaper system, and a molecular docking system. In the docking system, let $F_s = \sum f(x_i), x_i$ is the position of atom i, and $f()$ is a molecular force field function (color plate 2) [Ouh-young 88].

Contact instability and the human arm

Ideally a computer controlled joystick/hand-controller can simulate any target dynamics. However, in practice almost all systems have *contact instability* problems near a wall (a hard surface). There are several reasons,

1. If a digital computer is used in simulation, sampling delay can make a stable system unstable.
2. If one doesn't have measured external force F_{ext}, and approximates it with a velocity derivative, noise and delay are introduced.
3. The two different locations for sensor and actuator cause an instability problem (the non-colocation problem). The dynamics of the link (for example, the lower-arm of the ARM) itself are usually not properly modeled: the link is not a point mass, but is actually a distributed mass [Colgate 89].

Of course, one can make the system stable by adding extra viscosity, or by reducing the stiffness of the simulated hard surface. The former makes the human feel resistance and sluggishness even in free space, whereas the latter makes the hard surface spongy.

To make the problem even more complicated, the system is far from linear. The human operator's own physical characteristics are involved in the feedback loop in exploring the virtual world, and he changes those parameters dynamically and radically. Lanman reported human elbow stiffness to vary from a minimum of about 1.4 N-m/rad to a maximum as high as 400 N-m/rad [Lanman 80]. Cannon and Zahalak's measurements showed that both the limb's natural frequency and damping ratio vary with muscle activation [Cannon82].

Theoretical analysis [Murray 88] showed that a second-order model with parameters varying with muscle activation and elbow angle was unable to reproduce experimental observations. A simplest competent characterization required a fourth-order model. A fifth-order model was used by Hannaford [Hannaford89].

Hogan has experimental data to show that a human arm can be accurately modeled as a passive object with constant impedance for periods up to 1.2 seconds [Hogan89]. That is, it takes that long to change muscle impedance, rather than the 200 ms neuromuscular response time one might have expected.

All these data make satisfactory hard-surface simulation unlikely. But here is good news. First, multiple-sensory illusions (vision, force, sound) reinforce each other. Second, even though the system may become unstable during the simulation, the human operator can compensate or avoid it.

Analysis

What will be the behavior of the spring-mass system when a human arm is combined with it? Assume the human arm can be modeled by a second-order system with mass Mh, stiffness Kh, and viscosity Bh. If the arm is not generating forces, the system dynamics equation becomes

$$(Mh + m)a + (Bh + b)v + (Kh + k) * (x - x0) = 0 \quad (7)$$

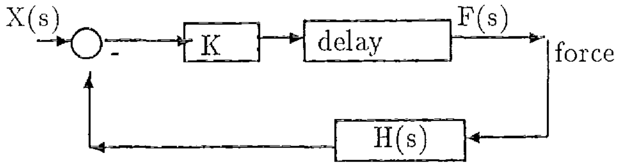
where a = acceleration, v = velocity, x = position, $x0$ = initial position, m = mass of joystick, b = viscosity of joystick, and k = stiffness of a virtual spring. The natural frequency f of the system is given by $2\pi f = \sqrt{(Kh + k)/(Mh + m)}$.

Delayed analog analysis

One way to predict the dynamics of Eq. 7 is to use analog control theory, and add a time-delay component in the feedback loop (Figure 6). We give a simple analysis of a spring and mass system (mass M, spring constant K, and viscosity B), with the human arm not included at first. Even though the delay (e^{-sT}, T is the delay in seconds) is introduced, this is still an analog controller. We use this model to get some insights before going on to the complicated digital controller.

Assuming that the product of the delay T and the natural frequency s is small, say less than 0.1, then e^{-sT} can be approximated by second-order Taylor series expansion.

$$e^{-sT} = 1 - sT + 1/2s^2T^2 \quad (8)$$



where K = spring constant,
$H(s) = 1/(Ms^2 + Bs)$, M: mass, B:viscosity,
delay = e^{sT},T: sampling period

Figure 6: An analog system with delay T

The transfer function between output force and input position becomes

$$\begin{aligned} F(s)/X(s) &= Ke^{-sT}/[1 + KH(s)e^{-sT}] \\ &= (Ms^2 + Bs)Ke^{-sT}/[K + (M + \\ &\quad 1/2KT^2)s^2 + (B - KT)s] \end{aligned}$$

By the Nyquist stability criterion, if one of the poles of F(s)/X(s) is located on the right-half of the s-plane, the system is unstable. The poles of F(s)/X(s) are equal to zeros of X(s), and are given as

$$[(KT - B) \pm \sqrt{(B - KT^2 - 4(M + 1/2KT^2))}]/(2(M + 1/2KT^2))$$

The system becomes unstable if $KT - B > 0$, where K is stiffness of the system, T is delay, and B is viscosity. This is an approximate solution. There is a constant C involved in this relation, i.e., T > C*B/K, and C is shown to be approximately 2 in more detailed discrete simulations.

When the human arm is combined with the system, let K = Kh + k, M = Mh + m, B = Bh + b, and $T < C*B/K$ still holds (see notations in Eq. 7).

A true discrete analysis

With a discrete model the solution is not in closed form, and we have to use numerical simulation to get insights from it.

Doing so we made the following observations. Let T^* be the maximum sampling period that makes the system stable.

1. T^* is linearly related to 1/K, where K is spring constant.
2. T^* is linearly related to viscosity B over a wide range, and then becomes nonlinear (when $B > 16$ N-sec/m) .
3. T^* is actually not related to spring mass M when the mass is over a threshold (0.02 Kg).

To understand this solution, suppose B (the viscosity) is small, as in many virtual world simulations, and K (the spring constant) is big, then the system delay T can easily be bigger than 2*B/K. A typical example would be B = 1.17 N-sec/m (joystick), a strong spring K = 400 N/m, and T > 2*B/K = 5.9 ms can cause instability. This places a severe restriction on the force fields that can be simulated by a slow update-rate system.

Similarly, if one simulates "stirring a rod in a tank of viscous oil" by $F_s = Bv$, where v is the joystick velocity, B is the desired viscosity, the stability condition is $T^* < C1 * M/B$, and $C1 = 2$.

Hard surface simulation with low sampling frequency

The following is an interesting experiment constructed to test the results from above discrete analysis.

> Procedure: In hard-surface simulation, let the system first run at 1000 Hz, then run at 250 Hz, then run at 100 Hz. Running at 100 Hz, when one bumps into the hard surface, the program increases viscosity to three times of human arm viscosity (5 N-sec/m), i.e., Bs = 15 N-sec/m.
>
> Results: At 1000 Hz, the system is stable; at 250 Hz, the system is unstable; at 100 Hz, the system is stable again. At 100 Hz, the subjects "feels" the same hard surface as if the system was running at 1000 Hz.

Parameters in this experiment.

Ks	Kh	Bs	Bh	Bs (within hard-surface)
2773	400	1.17	5	15

Mh (in Kg), Kh (in N/m), Bh (in N-sec/m) are human arm mass, stiffness, and viscosity; Ms (in Kg), Ks (in N/m), Bs (in N-sec/m) are joystick mass, stiffness, and viscosity.

Explanation: in order to be stable, $T < 2 * (Bs + Bh)/(Ks + Kh)$. The system is unstable in hard surface simulation at 250 Hz, since $(5 + 1.17) * 2/(2773 + 400) =$ 3.84 ms = 260 Hz. If the program increases Bs from 1.2 to 15 N-sec/m, three times of human arm viscosity, even the lower sampling rate (100 Hz) makes the system stable, since $(5 + 15) * 2/(2773 + 400) = 12.6$ ms = 79 Hz.

In our experiments, the subjects did not feel the viscosity difference, however, it helps tremendously in reducing the required sampling frequency

This was a very useful observation, and it shed light on other implementations. Possible conditions when the viscosity can be added without the loss of performance (in terms of human feeling) are:

1. within the hard-surface, which needs geometry information.
2. in any region where the equivalent stiffness is above a threshold (which causes instability at the given sampling rate). This can be implemented as a simple threshold function: if $2B/K > T$, let $B_{new} > TK/2$.

Hard surface simulation with two different hand motions

Procedure: use the joystick to bump into a hard surface, which is simulated by a spring with stiffness 2773 N/m, with sampling period at 2.8 ms.

Results: the tangential motion (side-to-side) is always unstable, but the radial motion (forward-backward) is always stable for all thirteen subjects (graduate students in the graphics laboratory).

Parameters used in this experiment.

Ks	Kh	Bs	Bh (tangential)	Bh(radial)
2773	400	1.1	3	10

Explanation: there is viscosity difference between radial and tangential motion. The stable condition is $T < 2 * (Bs + Bh)/(Ks + Kh)$. In the case of tangential motion, the system is unstable since 2*(1.1+3)/(2773+400) = 2.6 ms is smaller than the required sampling period of 2.8 ms. However, in the case of radial motion, it is stable, since 2*(10+1.1)/(2773+400) = 7.0 ms is well above 2.8ms.

Two puzzles about the behavior of human arms

We encountered two puzzles during the study of force display. First, how can the normal human be stable, even though the neural-muscular response time is around 200 ms? The puzzle was raised when the joystick we used had a sampling frequency of more than 30 Hz and still could easily be unstable. Is it because the human arm has a better way to compensate the system dynamics? The other puzzle is that even though the joystick sampling frequency was increased from 500 Hz to 1000 Hz, the human arm could still feel the difference in some cases.

In drama and literature, the human arm has been portrayed as the wings of a swan, the fists of a bear, the hammer that strikes the bell, and a piece of iron in a warrior. These magic tasks of human arms were created by illusions that looked realistic to human eyes.

Why is human arm always stable for a healthy person? If we assume that the human arm is implemented by a digital controller, the sampling period T must be smaller than 2*Mh/Kh in order to be stable. Typical values of Mh = 0.8 Kg and Kh = 500 N/m show that T must be smaller than 3.2 ms! Obviously this is not a correct model, since the known human neural-muscular response time is much bigger than 1.6 ms, and is around 200 ms.

Hogan coined a term *digitally supervised analog control* [Hogan87]. The idea is that an analog controller can eliminate sampling problems, at the same time allowing some control parameters to be updated by a digital computer infrequently and asynchronously.

Similarly, here we can think of a human arm as an analog controller supervised by the mind. But this mechanism is not perfect. Suppose the human arm wants to act like a piece of paper floating in the air, or an iron with big mass, the *digitally supervised analog control* is simply inadequate. The reason is that even though the human arm can sense the external force and change the muscular force, stiffness, and viscosity, the time delay is too big to make the arm act like a paper or an iron. Try the task of letting your hand behave like a piece of paper encountering a striking stick. Although one can see the coming stick by its trajectory, and feel its contact with the skin, it is impossible for one to make one's hand act like a piece of paper.

The second puzzle is that even though the joystick sampling frequency is increased from 500 Hz to 1000 Hz, the human arm can still feel the difference in some cases. Considering that human neural-muscular response time is about 200ms, this phenomenon is hard to explain at first. Our explanation to this puzzle is that although the joystick is running at 500 Hz, it may be unstable at that frequency when it is stable at 1000 Hz. The vibrations caused by this instability can be sensed by human hand, since there are skin sensors tuned as high as 400 Hz (sensitive to a range from 2 Hz to well over 500 Hz) [Sherrick86]. If the system is stable under both sampling rates (500 Hz and 1000 Hz), we observe that there are no gross differences in force perception in a few simulations in the Sandpaper environment. We hypothesize that the stability of a simulation is a major criterion in differentiating between them.

Conclusion

We did not use all the theories in designing our first systems. When problems came one by one, we realized that an analysis would be useful. The analysis helped us understand and improve the performance of our current systems, and we believe it will also contribute to the design of new force display systems.

Acknowledgements

The work reported in Part I was primarily done by Minsky and Steele, the work in Part II by Ouh-young.

Part I. We would like to thank David Banks, Susan Lederman, Kah-Chan Low, Doug Milliken, Ken Perlin, Alex Pentland and David Heeger, and Steve Strassmann. Thanks to Barbara Bowen for support and encouragement. Photograph by Bo Strain.

This work was supported by Apple Computer and the University of North Carolina at Chapel Hill Department of Computer Science.

Part II. This work was supported by the NIH Division of Research Resources, Grant RR-02170.

References

[Atkinson77] Atkinson, W.D., Bond, K.E., Tribble, G.L. III, and Wilson, K.R., "Computing with Feeling," Comput. and Graphics, Vol. 2, pp. 97-103, Pergamon Press, 1977.

[Batter72] Batter, J. J. and Brooks, F. P., Jr. GROPE-1: "A Computer Display to the Sense of Feel." Proc. IFIP 1972, 759-763.

[Brooks88] Frederick P. Brooks, Jr., "Grasping Reality Through Illusion: Interactive Graphics Serving Science," Invited keynote address at the Fifth Conf. on Computers and Human Interaction, Washington, D.C., May 17, 1988. Published in CHI'88 Proceedings, May 1988, 1-11. Addison wesley, 1988.

[Cannon82] Cannon, S. C. and Zahalak, G. I.,"The Mechanical Behavior of Active Human Skeletal Muscle in Small Oscillations," *Journal of Biomechanics*, Vol. 15,pp. 111-121, 1982.

[Colgate89] Colgate, E., and Hogan, N., "An Analysis of Contact Instability in Terms of Passive Physical Equivalents," *Proceedings of IEEE Robotics and Automation Conference 89*, pp. 404-409, Scottsdale, Arizona, 1989.

[Hannaford89] Hannaford B., "A Design Framework for Teleoperators with Kinesthetic Feedback," *IEEE Transactions on Robotics and Automation*, Vol. 5, NO. 4, August 1989.

[Hogan87] Hogan N., "Stable Execution of Contact Tasks Using Impedance Control," *Proceedings of IEEE Robotics and Automation Conference 87*, pp 1047-1054, Raleigh, NC, 1987.

[Hogan89] Hogan N., "Controlling Impedance at the Man/Machine Interface," *Proceedings of IEEE Robotics and Automation Conference 89*, pp. 1626-1631, Scottsdale, Arizona, 1989.

[Katz25] Katz, D., trans. and ed. by Lester Krueger, *The World of Touch*, Lawrence Erlbaum, 1989

[Lederman72] Lederman, S.J., and Taylor, M.M., "Fingertip force, surface geometry, and the perception of roughness by active touch," *Perception and Psychophysics*, Vol. 12, No. 5, 1972, pp. 401-408.

[Lederman87] Lederman, S.J., and Klatzky, R.L., "Hand movements: A Window into Haptic Object Recognition," *Cognitive Psychology*, Vol. 19, No. 3, 1987, pp. 342-368.

[Murray88] Murray, W.R.,"Essential Factors in Modeling the Modulation of Impedance about the Human Elbow," Ph.D dissertation, Department of Mechanical Engineering, M.I.T, 1988.

[Noll72] Noll, A.M., "Man-Machine Tactile Communication," J. Soc. Inform. Dis., July 1972.

[Ouh-young88] Ouh-young, M., Pique, M, Hughes, J., Srinivasan, N., and Brooks, F.P., Jr. "Using a Manipulator For Force Display In Molecular Docking,"*IEEE Robotics and Automation Conference*, Proceedings, Vol 3, pp 1824-1829, Philadelphia, April 1988.

[Pentland84] Pentland, Alex, "Fractal-Based Description of Natural Scenes," *IEEE Trans. Pattern Anal. Machine Intell.*, Vol. PAMI-6, No. 6, Nov. 1984, pp. 661-674.

[Perlin85] Perlin, K., "An Image Synthesizer," *Computer Graphics*, Vol. 19, No. 3, July 1985, pp. 287-296.

[Russo90] Russo, M., M.S. Thesis, MIT Dept. of Mech. E., in preparation.

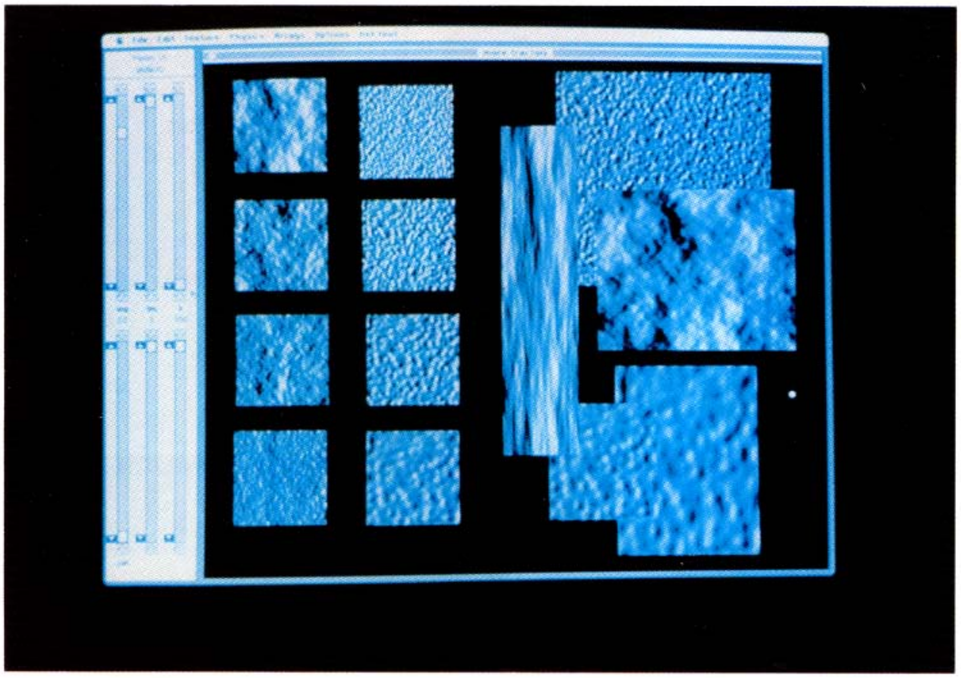
[Sherrick86] Sherrick, C.E., and Cholewiak, R.W., "Cutaneous Sensitivity," in Handbook of Perception and Human Performance. Eds.: K. Boff, L Kaufman, and J. Thomas. John Willy and Sons, N.Y., 1986.

[Smith88] M. J. Smith, "Tactile Interface for Three-Dimensional Computer-Simulated Environments: Experimentation and the Design of a Brake-Motor Device," M.S. Thesis, Mechanical Engineering Dept., Massachusetts Institute of Technology, 1988.

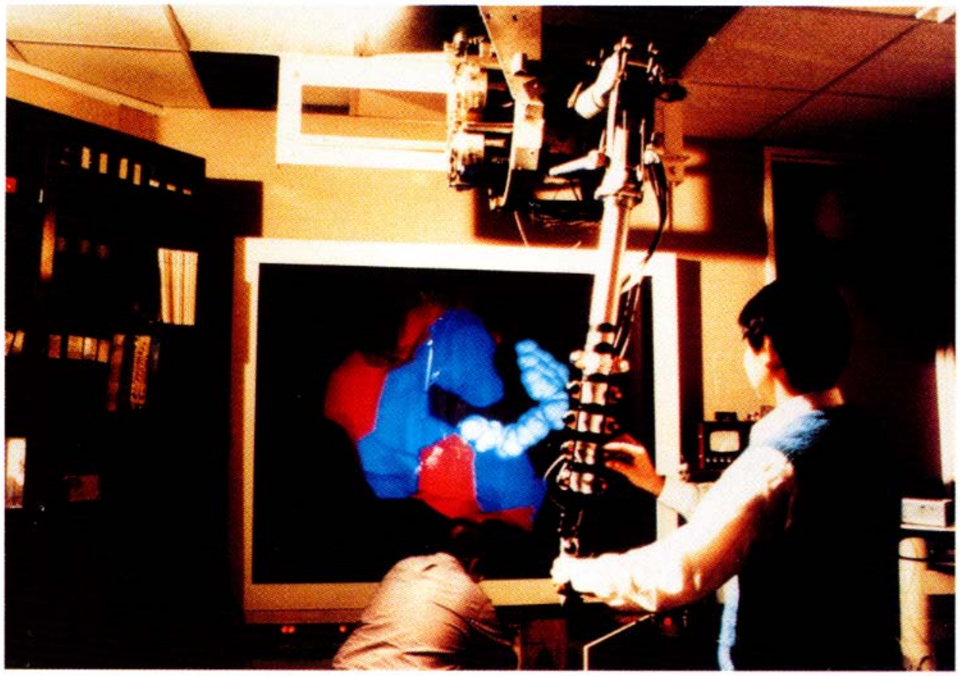
[Sutherland65] Sutherland, I.E. "The Ultimate Display," IFIP Vol.2, p. 506, 1965

Color images for this paper can be found in the color plate section.

Minsky, Ouh-young, Steele, Brooks and Behensky, "Feeling and Seeing: Issues in Force Display".



Color plate 1: The *Sandpaper* environment screen display. The patches have textures, animated objects, or materials in them. The user can feel these when holding and moving the force feedback joystick. The sliders on the left adjust the way each patch feels.



Color plate 2: The molecular docking system uses a master station of a remote manipulator system (ARM) as a 6-D force and torque interface. The purpose is to find the best fit of a drug inside a receptor molecule. The drug in white is methotrexate, an anti-cancer drug; the receptor in blue and red is dihydrofolate reductase, an enzyme.