

Interactive Exploration of Remote Objects Using a Haptic-VR Interface

John E. Lloyd and Dinesh K. Pai

Dept. of Computer Science, University of British Columbia
Vancouver, B.C., Canada

Abstract. We describe an interactive interface for exploring and acquiring reality-based models of physical objects. A user interacts with a graphical/haptic display to specify exploration operations on a model of the object (poking, rubbing along a path). These operations are then performed on the real object by a remote robotic sensing system, in order to measure certain requested physical properties (roughness, friction, acoustic response). Measured physical data is then parameterized and mapped back onto an area of the model in the haptic display, where it can be experienced by the user. If desired, the user can then extend this data onto other parts of the object model, by applying it to regions and features defined using the haptic display.

1 Introduction

We're interested in how a robot and a human can collaborate to explore and object and build multi-sensory models of an object. This type of remote exploration has many applications in telescience, where expensive robotic instruments such as ACME (the UBC Active Measurement Facility) [1] can be shared among geographically distributed scientists. The use of haptic technology for the remote exploration of rocks and geological surfaces has been discussed in [2]. Other applications include undersea and space robots, search and rescue robots, and home monitoring robots. In all these cases, it is important to provide natural, intuitive interfaces to robotic exploration without any programming.

In this paper, we describe a haptic-VR interface that allows a user to interactively explore and acquire physical models (such as contact roughness, friction, and acoustic response) for an object located at a remote site. The interface presents the user with a model of the object with which they can interact in order to request a particular measurement or modeling operation. In particular, the user can explore the "feel" of an object, and focus the modeling onto specific regions of interest. No programming is required.

Part of our motivation for this work is to study new ways in which 3D VR environments can be used to specify tasks and intention to the computer, in the same way that standard graphical user interfaces (GUIs) are now used universally for 2D applications such as graphic design. Other interesting work along these lines includes systems for artistic painting [3] and 3D modeling and sculpting [4].

2 A “Paint” Program for Physical Textures

Physical models of the sort that we are acquiring (deformation, roughness, acoustic response) are usually registered onto a surface mesh representation of the object, in much the same way that visual appearance can be mapped onto an object’s surface in the form of a texture map. For example, Fig. 1 shows

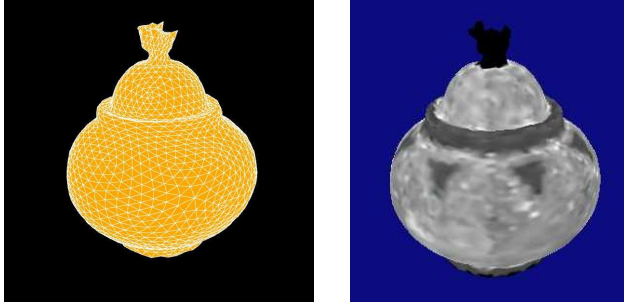


Fig. 1. A mesh representation of a clay vase (left), and a false color image of the vase’s friction (right)

a map of the surface friction of a clay vase that was measured using ACME. For purposes of this paper, we will refer to physical properties mapped onto an object’s surface as a *physical texture*.

In previously reported work, we described how the ACME system can be used to automatically acquire such models [5]. However, this process requires the user to write small programs, and do some form of exhaustive measurement on the object. For example, the friction map in Fig. 1 took several hours to complete. In the previous ACME measurement framework, it was also hard to direct attention to aspects and features of an object that mattered most from a perceptual viewpoint.

In this paper, we describe an interactive system in which the user directly guides object exploration and model acquisition using a haptic-VR interface. Putting the user in the loop overcomes the shortcomings described above: models can be acquired quickly, with attention directed toward the most important and salient features.

Model acquisition proceeds according to the following scheme:

- **Specification.** A user specifies what sort of data is to be measured, along with a tactile operation suited to obtaining this data (for instance, pressing the object at a certain point, or stroking it along a prescribed path).
- **Measurement.** The request is transmitted to the ACME system, which performs the operation and measures the requested data. The data is then parameterized and may also be processed to locate important features.

- **Experience.** The parameterized data is mapped back onto a user-specified region of the model, where the haptic interface can be used to experience the results of the measurement: the roughness of a surface, how it sounds when hit or scraped, etc.
- **Extrapolation.** Data can then be applied to other regions and features of the object defined by the user.

The last item is a key feature of the system. The user can define regions and features on the object (as presently implemented, this includes sub-areas of the surface mesh, and curves defined on the mesh) and “paint” them with data acquired by ACME. In this way, a small amount of measured data can be used to describe a large part of the object’s surface.

3 System Description

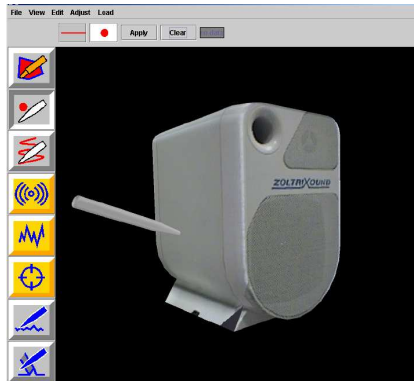


Fig. 2. System interface display

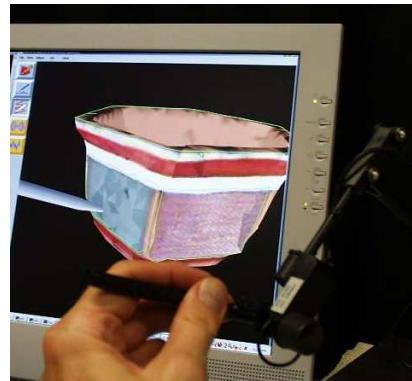


Fig. 3. System interface with the PHANToM Haptic device

3.1 User interface

It is assumed that a surface mesh representation (possibly texture mapped) is available for the object to be explored. This forms the basis of the object model, and is displayed in a viewing window of the user interface display (Fig. 2). Along the left side of the display are a number of buttons which can be used to select various *tools* with which the user can examine the current state of the model, or specify exploration operations to perform on the object. The tools are moved about on the object model via a SensAble Technologies PHANToM (Fig. 3).

Currently implemented tools include:

- A *marker* tool for defining area regions on the object’s surface.
- A *point* tool for placing points on the object to be used in point-oriented exploration operations (such as pinging or poking the object in a specific location).
- A *path* tool for drawing paths on the object for path-oriented explorations.
- A *probe* tool to examine the current state of the model (i.e., its haptic feel and acoustic properties).
- A *feature* tool to select point features that have been detected by the system and extrapolate them into large-scale features defined by curves on the object’s surface (Sec. 5.2).

Once exploration paths or points have been placed on the object’s surface, special buttons are used to request various forms of measurement data (within the current implementation, these include friction/roughness, sound, and positional registration data).

3.2 Remote measurement facility



Fig. 4. Tri-headed sensing tool, including (clockwise from top) solenoid-driven sound effector (for acoustic impulse response), contact-microphone probe (for scraping sounds), and a friction/roughness probe. The force sensor is located directly behind the sensing tool

Exploration requests are then transmitted to, and executed by, the ACME system [5] and the results are transmitted back to the interface and registered onto the mesh. Most requests require the system to contact the object (poke it at a point, stroke it along a path), and these requests are carried out by a 6 DOF Puma260 manipulator equipped with a wrist-mounted force/torque sensor and capable of active compliance operations. A special tri-headed sensing tool is mounted on the end of the manipulator (Fig. 4).

This ability to execute small command sequences autonomously at a remote site and transmit the results back to a model-based interface gives the system a teleprogramming/telesensing structure, similar in concept to those described in [6–8]. This makes it independent of both communication delays, as well as operational delays (it can take the robotic system longer to execute a request than it can take to specify it).

3.3 Data models

At present, the system measures surface roughness, friction, and sound properties. The parametric models which are used to describe this data have been described elsewhere and so will only be summarized here.

Sound is modeled as a combination of acoustic impulse response (represented using a modal resonance model) plus the excitation due to scraping/rolling contact. The resonance model is composed of several dozen decaying sinusoidal modes, while the scraping/rolling contact excitation consists of a short sample of the high frequency forces generated in response to such contact, which are then fed into the resonance model. The high frequency force sample is obtained from a sound sample obtained while scraping the object with a contact microphone. Details are given in [9].

Friction is modeled using conventional Coulomb friction and is measured (at a particular point on the object’s surface) from the forces observed while stroking the surface point in two different directions [10].

Roughness is modeled as a local variation in friction, which makes it both easy to measure and to render haptically. The variation can usually be characterized by an auto-regressive model driven by white noise, which is a compact representation requiring only a dozen or so parameters [10]. At present, the roughness model is isotropic.

Both friction and isotropic roughness are modeled on the object itself by a simple “friction map,” or “haptic texture,” which assigns friction values to each point on the surface mesh in a manner completely analogous to visual texture mapping.

4 Measuring and Applying Physical Textures

The most basic use of the system is for the user to apply different physical textures to different parts of the object. Following the scheme described in Sec. 2, this is typically done as follows:

4.1 Specifying an exploration

The user draws an exploration path on the object using the path tool (Fig. 5, left). For measuring surface roughness, this path can be arbitrary, while for measuring sounds the path should be straight (since the contact microphone probe works best while traveling quickly in a straight line). Constraining mechanisms in the interface facilitate the drawing of different types of paths, including zig-zagging paths for tracking grooves and similar features (Sec. 5.2).



Fig. 5. Specifying a path (left), measuring data (middle), and defining a region (right)

4.2 Measuring data

A sound or friction/roughness measurement is requested. For sound requests, the manipulator scrapes the object along the path using the contact microphone probe, and then “pings” the object using the sound effector (Fig. 4). A sound model is generated from the recorded sounds of both events and returned to the user interface. For friction/roughness requests, the manipulator performs a constant-force compliant motion along the path in two different directions (Fig. 5, middle), using the friction/roughness probe, and the resulting forces are used to determine the friction at different points along the path. A model of the mean friction and overall roughness along the entire path is computed and returned to the user interface.

4.3 Experiencing the result

Using the marker tool, the user defines a small region on the object in the area of interest (Fig. 5, right), to which the returned model data is applied. The selected region of the model can then be examined using the haptic display, feeling the friction/roughness model directly, or listening to the sounds generated when the object is scraped or tapped.

4.4 Extrapolating results onto the object

If satisfied with the result, the user can retain the applied physical texture, and/or apply it to other regions defined on the object. In this way, the physical textures acquired from the object can be thought of as forming a palette of textures which can be used to “paint” other parts of the object.

5 Detecting Features and Changes in Physical Texture

In defining a region of the object on which to apply a particular physical texture, the user is typically guided by (a) the visual appearance of the object,

in terms of either its shape or visual texture, or (b) personal experience with objects of similar type (if one knows that a particular artifact feels the same all over, then a friction/roughness sample from one part of the object is likely to suffice for the entire thing).

On the other hand, situations may arise where the object is not well known to the user, or the physical texture has features which are not apparent from the available shape and visual texture information. Of particular importance are discontinuities or anomalies in the physical texture. The system should either detect such situations, or make it easy for the user to do so.

5.1 Visualization of physical texture

A direct way to detect changes in the physical texture is to map a visual representation of it directly onto the object, making it easy for the user to see patterns that may not be apparent by feeling it directly.

It is particularly easy to do this for our isotropic friction/roughness texture, which consists simply of friction defined as a scalar function over the object's surface. This can be easily mapped into a false-color image texture on the object's surface, in the same manner as that user to generate Fig. 1.

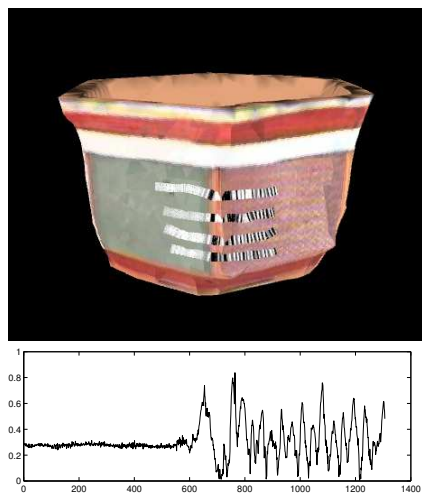


Fig. 6. Texture mapped image of an object, overlaid with false color images of the friction encountered along four exploration paths. The region on the paths' left side is a smooth piece of rubber, while the region on the paths' right side is a piece of wire meshing. The change in roughness between the rubber and the wire mesh is plainly visible in the false color image. A plot of one of the actual friction profiles is shown below

More specifically, the system will, if requested, display a false color image of the friction measured along each point of an exploration path (Fig. 6).

5.2 Detection of bumps and feature extrapolation

It is also possible for the system to preprocess measured physical data, detecting features and anomalies which are then be mapped back onto the model for further consideration by the user.

The system currently implements a bump detector, which works by looking for sharp peaks in the friction measured along a path (see [11] for another approach to haptic feature detection). Points where bumps occur are then displayed on the model. If the user suspects that these points belong to a larger-scale feature such as a groove, he or she can create a curve corresponding to this groove on the model's surface (using a curve fit to the detected bump points and/or explicit drawing). Information from the bumps themselves (width, intensity) can then be combined with this curve to create a modeled groove on the object's friction map (Section 3.3) that can be felt using the haptic display.

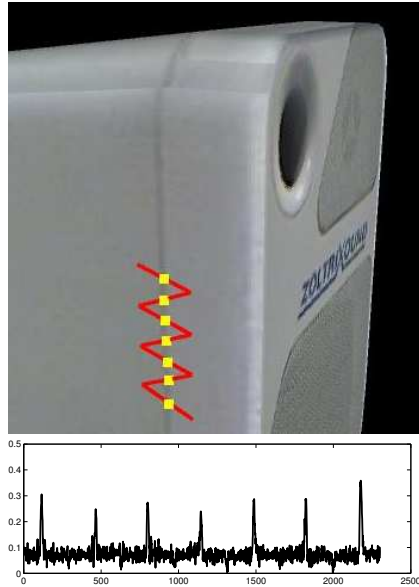


Fig. 7. Zig-zagging exploration path on the surface of an object (in this case, a computer speaker). The friction measured along the path is shown below. The plainly visible peaks in the friction correspond to bumps on the path, for which the corresponding points are displayed on top of the path

An example of this is shown in Fig. 7. Here, a special zig-zig exploration path was requested along a visual line on the object's surface to see if this actually corresponded to a groove. Peaks in the friction indicated that it did. A graphical line was later fit to the detected bump points, and extended by the user along the visible line, in order to define a modeled groove on the object.

Creating a groove by interpolating and extending a small number of detected bumps is a specific example of what we mean by *extrapolation* (Sec. 2). It can be thought of as a one-dimensional version of creating a region on the surface and applying measured physical data to it.

6 Overall Results and Future Work

By working with this system on several objects, we have seen that it is fairly fast and easy to use. For example, modeling the computer speaker (Figs. 2 and 7) took about 5 minutes. This included defining about 6 regions on the object, collecting sound and roughness data for each, and exploring and extrapolating the two grooves that surround the speaker at the front and back.

In doing this work, we have determined a number of issues that should be addressed in the future.

- **Registration of visual and physical textures.** If the visual texture mapping of an object is not accurate, there may be a difference between where an artifact appears visually and where it appears in the physical texture. The effect is disconcerting, and ways of handling this problem need to be developed.
- **Visualization of physical textures.** We believe that being able to visualize a physical texture, in addition to just feeling it, provides an excellent means of augmenting the user’s understanding of an object, and further work should be done in this area.
- **Pre-segmentation of textures.** It would be useful for the system to automatically segment the physical and visual texture information in order to suggest region boundaries or other features to the user (this would be an extension of the feature detection described in section 5.2). This information could then be used, combined, or modified at the user’s discretion. For example, roughness data could be segmented by building on existing techniques for detecting changes in signals (e.g., [12]).
- **Extensions.** A number of extensions should be easy to implement, such as the detection of deformability. Also, while performing explorations, a large amount of geometric information is gathered which could be used to improve the object’s underlying geometric model (i.e., the surface mesh).

7 Conclusion

We have implemented an interactive system for modeling and exploring the physical properties of objects, built on top of the ACME system for acquiring reality-based physical models. A haptic display is used to allow the user to interact with the object model and specify exploration requests to be executed by ACME. The system improves on the previous version of ACME by putting a user directly in charge of the modeling operation. This allows model acquisition to be focused directly on the most critical areas and features of an object, and also permits the user to apply (or extrapolate) a small number of data measurements onto user-defined regions and features. In this way, the program resembles a “paint” program where physical textures, acquired by

ACME, are applied to the object under user control. This allows models to be acquired easily and quickly, greatly minimizing the amount of physical measurement required.

References

1. D.K. Pai, J. Lang, J.E. Lloyd, and J.L. Richmond, "Reality-Based Modeling with ACME: a Progress Report", *7th International Symposium on Experimental Robotics*, Honolulu, Hawaii, December 2000.
2. A. M. Okamura, M. A. Costa, M. L. Turner, C. Richard, and M. R. Cutkosky, "Haptic Surface Exploration," *Proc. of Sixth Intl. Symp. on Experimental Robotics*, 1999.
3. B. Baxter, V. Scheib, M.C. Lin and D. Manocha, "DAB: Interactive Haptic Painting with 3D Virtual Brushes", *Proceedings of ACM Siggraph*, Los Angeles, California, August 2001.
4. S.A. Ehmman, A.D. Gregory and M.C. Lin, "A Touch-Enabled System for Multiresolution Modeling and 3D Painting", *Journal of Visualization and Computer Animation*, 2001.
5. D.K. Pai and K. van den Doel, D.L. James, J. Lang, J.E. Lloyd, J.L. Richmond and S.H. Yau, "Scanning Physical Interaction Behavior of 3D Objects", *Proceedings of ACM Siggraph*, Los Angeles, California, August 2001.
6. J.E. Lloyd, J.S. Beis, D.K. Pai and D.G. Lowe, "Programming Contact Tasks Using a Reality-based Virtual Environment Integrated with Vision", *IEEE Transactions on Robotics and Automation*, June 1999, (vol RA-15, no. 3), pp. 423-434.
7. G. Hirzinger, B. Brunner, J. Dietrich and J. Heindl, "Sensor-Based Space Robotics - ROTEX and Its Telerobotic Features", *IEEE Transactions on Robotics and Automation*, October 1993, (vol RA-9, no. 5), pp. 649-663.
8. C. R. Sayers, *Operator Control of Telerobotic Systems for Real World Intervention*, Ph.D. Thesis, Department of Computer and Information Science, University of Pennsylvania, Philadelphia, Pennsylvania, 1995.
9. K. van den Doel, P.G. Kry and D.K. Pai, "FoleyAutomatic: Physically-based Sound Effects for Interactive Simulation and Animation", *Proceedings of ACM Siggraph*, Los Angeles, California, August 2001.
10. J.E. Lloyd and D.K. Pai, "Robotic Mapping of Friction and Roughness for Reality-based Modeling", *Proceedings of the IEEE International Conference on Robotics and Automation*, Seoul, Korea, May 2001.
11. A.M. Okamura and M.R. Cutkosky, "Feature Detection for Haptic Exploration with Robotic Fingers." *International Journal of Robotics Research*, Vol. 20, No. 12, pp. 925-938, 2001.
12. M. Basseville, "Detecting changes in signals and systems - a survey", *Automatica*, 1988, (vol 24, no. 3), pp. 309-320.