

Haptic Display of Contact Location

Katherine J. Kuchenbecker[†] William R. Provancher*
Günter Niemeyer[†] Mark R. Cutkosky*

[†]Telerobotics Lab and *Dexterous Manipulation Laboratory
Stanford University, Stanford, California
{katherine.kuchenbecker, gunter.niemeyer}@stanford.edu
{wil, cutkosky}@cdr.stanford.edu

Abstract

This work presents a new haptic device that integrates contact location feedback with grounded point-force display. The system consists of a thimble-based mechanism attached to the endpoint of a Phantom[®] robotic arm. Contact location is rendered using a small tactile element that moves along the length of the user's fingerpad. The Phantom[®] robot applies reaction forces to the user's finger through this tactile element, bringing it into and out of contact with the user as required. Force and contact location are thus displayed concurrently. During operation, the system continually adjusts the position of the contact element based on finger motion and expected or actual contact with the two-dimensional virtual environment. The finger is modeled as an arc segment, and the environment as a series of lines and arcs. Haptic rendering is driven by a virtual finger proxy, employing collision detection and collision anticipation algorithms. A series of human subject tests compared contact location feedback to standard force feedback. Subjects completed a contour following task in less time and with fewer failures when contact location information was available. The system's success indicates a simple yet promising new avenue for the design of haptic displays.

1 Introduction

Though touch is commonly regarded as a single sensory pathway, the ability to feel the world around us is truly a collection of layered sensations. When you manipulate an object in your hand, you can feel reaction forces, local pressure distributions, contact location, texture, temperature, and vibrations, as well as a kinesthetic awareness of your finger configuration. These sensations all work together to build a rich haptic image of the item you are holding.

Haptics research seeks to recreate this complex sense of

touch for users in virtual reality and telerobotics. Ideally, interacting with a virtual or remote environment would be just as simple and vivid as using a hand tool or your own fingers. Technology available today cannot yet meet this ambitious goal, so systems must be streamlined to contain only the information that is most important for the task at hand. Selecting the most salient feedback modes and rendering them with high fidelity can produce haptic displays that begin to resemble the ideal.

Force display has become the most prevalent haptic feedback modality, employed in such diverse applications as flight simulation, computer-aided design, and telerobotic surgery. The user manipulates a joystick, stylus, or thimble, and the mechanism applies corresponding forces to his or her hand. The human input is usually modeled as a point, mapped to the user's location in a virtual world or to the position of the end-effector of the remote robot. The force feedback vector is continually computed from the environment model or measured at the slave and displayed at the haptic device's endpoint.

Using a point-force display is equivalent to prodding the world with the tip of a stick. Sharp features are difficult to localize because you can touch only one part of the environment at a time. Yet when using your finger to explore objects like the edge of a table, you can quickly find the corner by feeling where it acts along your skin. The absence of this contact location feedback in standard haptic interfaces limits the user's sensitivity and dexterity during manipulation and complex exploration tasks.

Tactile displays, on the other hand, provide detailed feedback of local shape and pressure distribution on the user's fingertip. Accurate recreation of contact on a patch of skin requires a dense array of actuators, and each small element must provide high levels of force, velocity, and displacement. Most tactile displays are thus bench-top devices, with a small array of pins in a stationary frame, actuated via wires or tubes [1, 2, 6, 7]. The bulk and complexity of

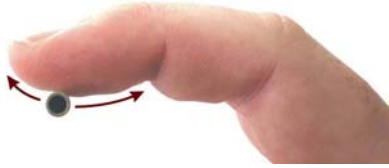


Figure 1. Contact location display concept: a tactile element moves along the fingertip to indicate the position of contact.

such devices all but precludes their use at the endpoint of a force-feedback system.

Alternatively, a display could render just the centroid of contact on each finger, rather than the entire contact profile. This strategy provides a simpler means of conveying important tactile information during haptic interactions. As illustrated in Fig. 1, a single contact element can traverse the surface of the finger in the proximal/distal direction as the location of contact with the virtual or remote object changes. This concept requires just one actuator to drive the tactile element to the desired location along the user's fingerpad. The simplicity of this approach facilitates its integration with traditional force feedback.

To investigate the merit of a hybrid tactile-haptic display, we developed a system that provides concurrent contact location and force feedback to the user. The device's effectiveness was previously evaluated in a series of human subject tests, as documented in [8, 9]. Results showed that users of the system could discern object curvature with a level of success similar to that of real manipulation. It was also found that users could discriminate between different types of virtual object motion, including rolling and anchored behaviors. Encouraged by the success of these initial investigations, we have since enhanced the system's hardware, interaction model, and controller to support general exploration of planar virtual environments.

The new contact location display system allows users to explore a two-dimensional virtual environment, feeling contact location and interaction forces simultaneously. The moving tactile element is housed in a thimble, which is worn on the index finger. The user moves his or her finger forward, backward, up, and down in a vertical plane and watches a side view of the interaction graphically on a nearby monitor. As described in Section 2, the system hardware uses a planar linkage to measure finger angle and position and regulate contact location and reaction forces. These commands are computed by a real-time model of the interaction between the fingertip and the virtual environment, as detailed in Section 3. The system's controller applies the forces and adjusts the position of the tactile element along the user's finger throughout the interaction, as discussed in

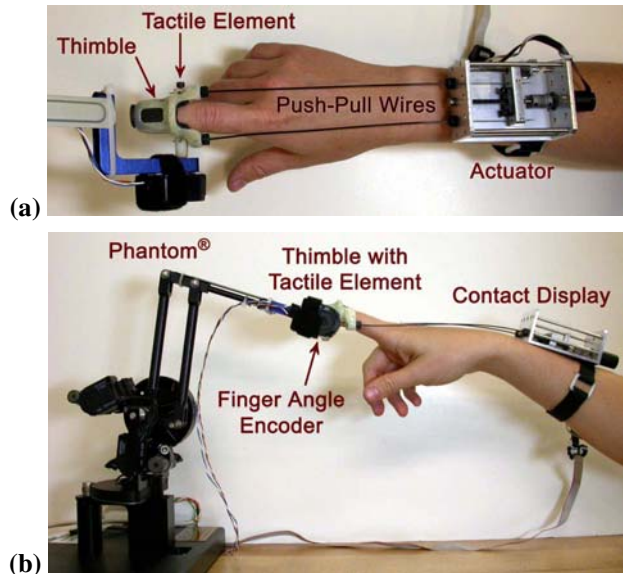


Figure 2. (a) Custom contact location display hardware. (b) System apparatus.

Section 4. These three system components work together to give the user the illusion of touching a two-dimensional contour, feeling the various features travel along the skin of his or her fingertip.

In order to evaluate the usefulness of contact location display, we performed a human subject experiment using the new system. As described in Section 5, users completed a contour following task under two test conditions: force feedback alone and force feedback with contact location display. The results from this study indicate that contact location significantly improves the user's ability to follow a contour smoothly, as presented in Section 6. Finally, conclusions and suggestions for future work appear in Section 7.

2 Hardware

The contact location display system combines custom hardware with a standard haptic feedback device to create a planar mechanism capable of rendering contact location and force feedback simultaneously. Modulation of contact location is achieved through a one-degree-of-freedom linear mechanism attached to the user's forearm and finger, as shown in Fig. 2(a). The tactile element is a small cylinder that is slightly sprung away from the fingertip to prevent contact in the absence of force feedback. Various tactile elements can be installed, including ones that rotate freely and others that remain fixed in orientation. The contact element can translate about 2.0 cm along the length of the thimble,

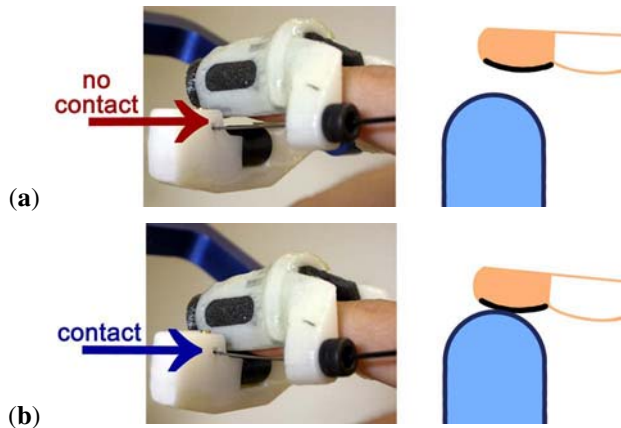


Figure 3. (a) Free-space motion creates no contact with the tactile element. (b) Touching a virtual object yields contact.

driven via two sheathed push-pull wires. A small DC motor actuates the wires via a leadscrew, continuously moving the tactile element to the appropriate location along the fingertip, as measured by the motor's encoder. Remotely locating this motor on the user's forearm reduces device inertia at the finger and minimizes the transmission of actuator vibrations to the user's fingertip receptors. A series of interchangeable thimbles was created using rapid-prototyping techniques to ensure a snug fit for a range of finger sizes.

As depicted in Fig. 2(b), the tactile element is attached to the endpoint of a desktop Phantom[®], a commercial device commonly used for point-force feedback [5]. The shoulder and elbow joints of this robotic arm allow user motion in a vertical plane, and a restoring torque at the base joint restricts finger movement to the center plane of the device's workspace. The encoders on the arm joints are used to compute endpoint position, and the motors act to apply forces to the user's finger through the contact element. An encoder mounted at the endpoint, coincident with the top of the tactile element, measures the orientation of the thimble's drive wires relative to the last link of the Phantom[®].

The tactile element is suspended underneath the fingertip by its two drive wires. When the user's virtual finger is in free space, the contact element does not touch the finger because no forces are applied. As illustrated in Fig. 3(a), there is a gap between the fingerpad and the cylinder in this situation. When the user comes into contact with a virtual object, the system moves the contact element to the correct location and applies a contact force at this point. This force pushes the suspended cylinder into the user's finger, as shown in Fig. 3(b), giving the user the impression of touching a virtual object. Such an arrangement creates a realistic sensation of making and breaking contact by stimulating ap-

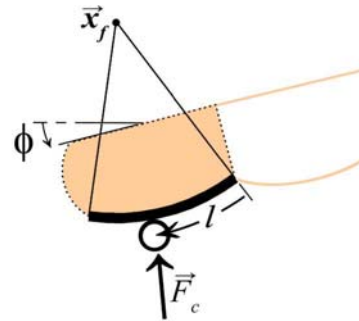


Figure 4. The user's finger is modeled as an arc segment with a radius of 2.6 cm. The rest of the finger is disregarded in collision detection.

propriate mechanoreceptors in the user's fingertip[10, 11]. A standard haptic interface can be transformed into a richer display by the simple addition of a linear positioning element, open-fingerpad thimble, and finger-angle sensor.

3 Interaction model

Combining the haptic display of contact location with force feedback requires a unique virtual interaction model. The system must treat the user's finger as an object, rather than a point, so that it can touch the environment at any location along its length. The position of contact along the finger continuously changes based on the user's motions and the geometry of the virtual environment. The tactile element must be positioned correctly while in contact with an object and when in free space, in order to adequately anticipate future contacts. The algorithm that drives this continual computation of contact location is based on simple models of the user's finger and the environment.

The interaction model for contact location display monitors the position and orientation of the user's finger, constantly updating a matching virtual finger. Following standard haptic display methods, this virtual finger acts as the user's proxy in the virtual environment [12]. It tracks the real finger's motions in free space but remains on the surface of the environment during contact. The contact location of the virtual finger drives the contact location display when the user touches the environment.

The system models the user's finger as an arc segment, corresponding to the surface of the distal fingerpad, as illustrated in Fig. 4. The arc segment can touch the environment along its curve or at either of its endpoints. The chosen arc segment has a radius of 2.6 cm, approximating the curvature observed in human subjects. It has an arc length of 2.0 cm,

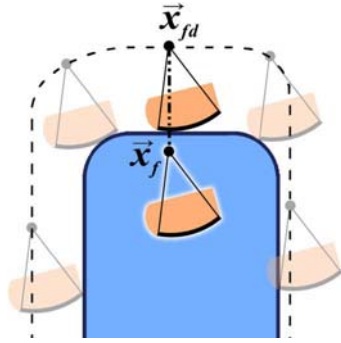


Figure 5. For a given finger orientation, tracing the arc segment along the environment boundary produces the configuration space contour, depicted with a dashed line. The desired position for the finger, \vec{x}_{fd} , is the history-dependent projection of the measured finger position, \vec{x}_f , onto this boundary.

equal to the length of travel of the contact display. The finger's configuration is described by the position of its center of curvature, \vec{x}_f , and its angle from horizontal, ϕ . Although the body of the finger is depicted in illustrations and on-screen graphics, the system does not currently consider the top, front, or back of the finger for collision detection.

As the user moves his or her finger around in the vertical plane, the interaction model must determine the appropriate contact force \vec{F}_c and contact location l , as defined in Fig. 4. The system calculates contact force based on the difference between real and virtual finger positions. The user feels a force pushing his or her finger toward the location of the virtual proxy on the environment's surface. Similarly, the system calculates contact location as the site of contact along the virtual finger's arc segment. In the event of multiple contact points, the system renders the centroid of contact along the arc segment.

The planar environment with which the finger interacts is modeled as a continuous series of arc segments and line segments, as shown for example in Fig. 5. An implicit representation was chosen to facilitate quick changes to the environment's geometry, as well as to simplify the calculations required during an interaction. A variety of layouts can be composed from these simple elements, creating a rich haptic environment for the user to explore.

3.1 Collision detection

During operation, the system detects collisions between the arc segment of the virtual finger and the composite contour of the environment. Performing such calculations on

a moving, tilting finger requires a careful examination of the geometry involved. Standard distance-based methods do not suffice because collisions also depend on the orientation of the arc segment. One way to address this problem is to treat it as a three-dimensional configuration space (c-space) with ϕ and the two components of \vec{x}_f as coordinates. In such an approach, each configuration would be mapped onto the virtual environment, delineating whether it results in a collision [4]. When the user moved his or her finger into an obstacle, the virtual finger would move to the closest free-space configuration, and the system would display forces and a torque to pull the user toward the virtual finger's configuration.

However, the contact location display mechanism cannot provide torque feedback on finger angle, so a modified approach is required. Finger angle must be treated as a driven coordinate; the angle of the virtual finger always tracks ϕ , the angle of the user's finger. For a given value of ϕ , the environmental constraint becomes a two-dimensional region into which the center of the finger's arc segment cannot travel. This c-space contour can be visualized by tracing the arc segment along the surface of the environment, as shown in Fig. 5. The boundary changes continuously with finger orientation and is computed by the system on the fly. Such a treatment transforms the collision detection problem into a two-dimensional interaction between a point and a region, which can be handled with standard approaches. In this approach, the system simply monitors the location of the finger and notes when it enters the present obstacle region in c-space.

Once the system has determined that the user is in contact with the environment, it computes the appropriate virtual finger location on the c-space boundary, as shown in Fig. 5. At each servo cycle, the system compares the presently measured finger position with the previously computed virtual finger position, determining which c-space element is now being contacted. The finger position \vec{x}_f is projected onto the selected boundary element to yield \vec{x}_{fd} , the desired position for the user's finger in the contact force controller. The system also uses the virtual finger's interaction with the environment to calculate l_d , the desired location of contact along the fingertip for the contact location controller.

3.2 Collision anticipation

When the user's finger is in free space, the device must predict and track the most likely point of contact so that the tactile element will be correctly positioned when the user touches the environment. This collision anticipation can be performed as an extension of the collision detection strategy discussed above. When the user is not in contact with the environment, the system identifies the c-space segment

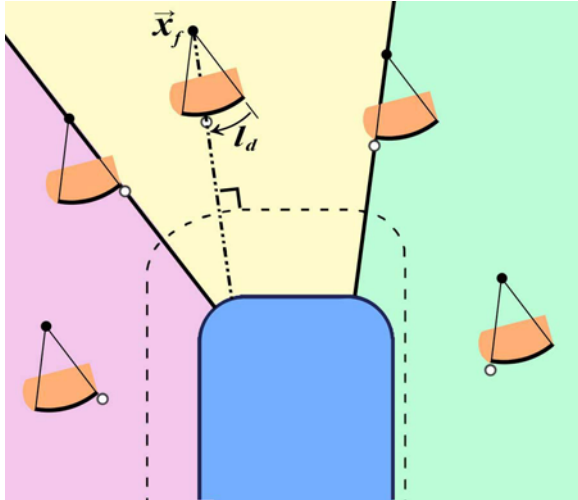


Figure 6. Free space is divided into zones for collision anticipation: the tactile element is driven to a point along the span of the finger's arc or to either of its two endpoints.

to which the finger is closest, as illustrated in Fig. 6. It then projects the finger position \vec{x}_f perpendicularly onto that c-space segment and calculates the contact location l_d that would result from such a collision. This tactic effectively divides the free space into zones based on the normal vectors of the boundary, as shown by the three regions in Fig. 6. Each zone corresponds to a certain type of environmental contact, involving either the arc or one of its two endpoints.

With convex environments, such a strategy creates a continuous mapping between finger position and contact location. Avoiding discontinuities is critical for smooth system operation because sudden device motion is disconcerting to the user during free space motion. Implementing more complex environments which include concavities will require a more sophisticated collision anticipation algorithm. The system will need to estimate the imminence of all the reachable contacts, average them together, and smooth the signal to prevent any discontinuities in contact location. The present c-space algorithm provides a framework for such improvements.

4 Controller

Coordinating the hardware with the virtual world during a haptic interaction requires precise control. The interaction model needs to monitor finger position, \vec{x}_f , and angle, ϕ , but the device does not directly sense these parameters. Similarly, the interaction model generates desired positions for the finger, \vec{x}_{fd} , and the tactile element, l_d , but the de-

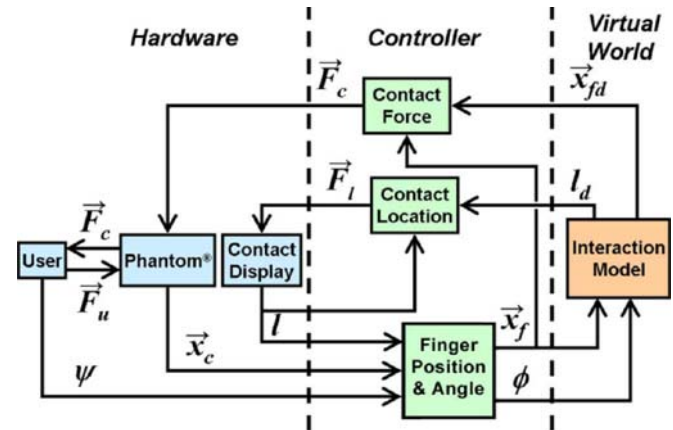


Figure 7. System Diagram. The controller connects the hardware to the virtual world, computing finger configuration, generating contact forces, and tracking contact location.

vice can output only forces, not positions. The controller links the system's two halves together with three components: finger configuration, contact force, and contact location, as illustrated in Fig. 7. The system hardware is connected to a computer running RTAI Linux, which performs a real-time servo loop at 1 kHz. During each cycle, the controller computes finger position and angle for the interaction model and then closes control loops around the desired positions for the finger and contact element.

4.1 Finger Configuration

The system cannot directly measure finger position and angle, but instead must construct these parameters from other signals. Forward kinematics are used to transform the device's four raw encoder readings to four coordinates of the mechanism's configuration: $[x_c \ y_c]^T$, ψ , and l . As illustrated in Fig. 8, \vec{x}_c is the Cartesian position of the top of the contact element and also the endpoint of the robotic arm, ψ is the angle of the drive wires relative to the last arm link, and l is the contact location along the finger's arc segment. The controller uses the finger model's geometry to determine the position, \vec{x}_f , and angle, ϕ , of the finger from these readings.

4.2 Contact force

Contact forces stem from differences between the user's finger position and the present location of the virtual finger. Because the angles of the virtual and real fingers are identical, the position error can be computed at either the arc center, \vec{x}_f , or the contact location, \vec{x}_c . The present interaction

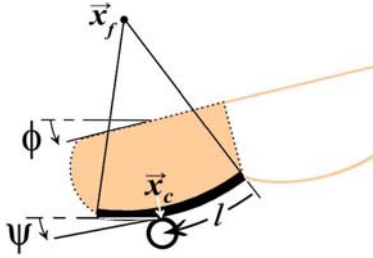


Figure 8. The geometry of the finger model allows the controller to determine ϕ and x_f from the sensed parameters ψ and x_c .

model simulates a simple stiffness, K_e , generating forces that are normal to the environment's surface as follows:

$$\vec{F}_c = K_e(\vec{x}_{fd} - \vec{x}_f) = K_e(\vec{x}_{cd} - \vec{x}_c) \quad (1)$$

This contact force is rendered using the shoulder and elbow motors of the robotic arm, and it acts on the user through the tactile element. At maximum current, the device can output about 1.5 N in any direction, which easily deflects the drive wires and pushes the contact cylinder into the user's finger. Future work on the interaction model could include simulated friction or other haptic cues in addition to stiffness.

4.3 Contact location

The system uses a position control loop on the contact display's linear degree of freedom to track the interaction model's specified contact location. Proportional and integral forces draw the tactile element to its desired location along the fingertip. Local derivative feedback is used to damp out the oscillations excited by the push-pull wires and the stiction of the leadscrew. The entire contact location control law is:

$$\vec{F}_l = (K_p + \frac{K_i}{s})(l_d - l) - K_d s l \quad (2)$$

where s is the Laplace operator. Because the leadscrew is non-backdrivable, the controller can force the tactile element to track its desired trajectory closely regardless of finger movements. The device can track 1.0 cm sinusoidal commands with a bandwidth of approximately 8 Hz. Contact position along the finger is rendered with a maximum error of 0.05 mm, which is imperceptible to humans [9]. This simple PID controller yields good performance and stable operation for the contact location display hardware, enabling its use in general exploration of planar haptic environments.

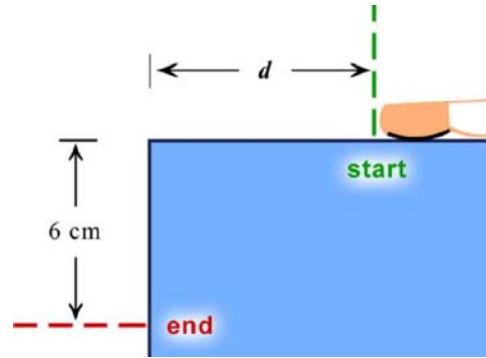


Figure 9. Subjects followed a virtual contour, trying to maintain contact throughout the interaction. Time was measured as the finger's arc segment traveled from the start to the finish line along the rectangular block.

5 Experiment

Humans use a variety of procedures to explore the environment around them with their highly sensitive fingertip receptors. Specifically, contour following provides information about the global shape of the object being touched [3]. Conventional haptic display systems provide only a single interaction point or sphere; following contours with such systems is difficult, especially when the surface has abrupt changes in direction or curvature, as in a table edge. Adding contact location display can improve the user's ability to perform these tasks by rendering the environment more realistically.

We constructed a simple virtual exploration task to investigate the possible benefits of contact location display when finding edges and following sharply changing contours. In this task we presented subjects with a rectangular block like that shown in Fig. 9. While blindfolded, they were asked to trace their finger forward along the top surface and find the corner. They were then instructed to tip their finger forward and move down the far edge of the virtual block without breaking contact with it. Subjects completed the task both with and without contact location display. In both cases the bottom of the finger was modeled as an arc segment, as described in Section 3, and force feedback was provided. In the trials where no contact location information was presented, the contact element was positioned statically in the center of its travel.

Subjects were presented with blocks of seven different lengths from 4 to 10 cm. Each of these values for d was presented approximately five times under each of the two test conditions, and the order in which the lengths were presented was randomized to reduce habituation. After a short

<i>Force Feedback Alone</i>			
<i>d</i>	Completion Time		Failure Rate
	Mean	Std. Dev.	Mean
4 cm	4.38 s	3.04 s	43.6%
5 cm	5.24 s	3.34 s	47.4%
6 cm	7.11 s	5.62 s	27.0%
7 cm	8.59 s	5.18 s	45.0%
8 cm	8.88 s	5.21 s	35.9%
9 cm	11.31 s	7.31 s	43.2%
10 cm	10.38 s	6.97 s	42.1%

<i>With Contact Location Display</i>			
<i>d</i>	Completion Time		Failure Rate
	Mean	Std. Dev.	Mean
4 cm	3.03 s	1.02 s	26.2%
5 cm	3.66 s	1.37 s	21.0%
6 cm	3.66 s	1.34 s	9.3%
7 cm	4.12 s	1.98 s	7.7%
8 cm	4.75 s	2.01 s	2.4%
9 cm	4.90 s	2.83 s	4.6%
10 cm	5.49 s	2.55 s	4.5%

Table 1. Pooled subject data for contour following performance with and without contact location display for the seven tested values of d . Average completion time and standard deviation are reported for successful trials only.

training period, subjects completed all trials for one test condition. They were then given a short break before beginning the second session. The order of the two test conditions was balanced among test subjects to reduce the effects of learning and fatigue. Each subject took approximately 30 minutes to complete the experiment, performing a total of about 70 trials.

Test results shown herein represent data from seven subjects ranging in age from 24 to 30, including both novice and experienced haptic users. The system automatically recorded completion time and trial success for each run. The timer started when the user's finger crossed a horizontal threshold at a distance of d from the edge and ended when the finger crossed a vertical threshold 6 cm from the top of the block, as shown in Fig. 9. A success was recorded only when the subject remained in contact with the block during the entire trial.

6 Results

Results from the human subject experiment are presented in Table 1 and Fig. 10. Table 1 gives the mean completion time and standard deviation for the contour fol-

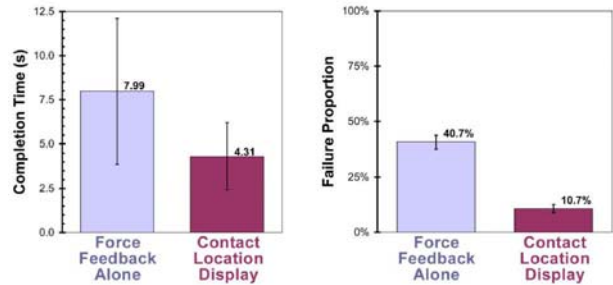


Figure 10. Experimental results pooled across subjects and trials. Average completion time and failure incidence both decrease with contact location feedback. Standard deviations are indicated by error bars.

lowing tests with and without the display of contact location. The data presented was pooled from all subjects, with completion times reported for successful trials only. The percentage of times subjects failed to complete the task by breaking contact with the environment is also listed for each trial type.

The mean completion time for trials under both test conditions scales roughly with block length, d . This trend indicates that subjects proceeded across the top of the block at an approximately constant speed, trying to detect the corner. The speed at which subjects proceeded differed between the two feedback conditions, though, indicating the adoption of different sensing strategies. Average completion time was 39.8% faster when subjects were provided with contact location information, as shown in Fig. 10. This finding is statistically significant, and the 99% confidence interval for the true average difference in completion times is (2.8693, 4.4871) seconds. Though completion times varied from subject to subject, all explored the contour with greater speed when contact location was available.

We also observed a significant decrease in the proportion of failed trials when users received contact location information. As indicated in Fig. 10, subjects were 30.0% less likely to break contact with the environment when provided with contact location feedback. The 99% confidence interval for the true average difference in probability of failure is (0.2949, 0.3047). Statistical analysis, based on a t test, shows the differences in both completion time and failure proportion to be significant at a 99% confidence level.

To complete the task successfully, a user must be able to sense that the edge of the block is imminent. Once the user has identified the corner, he or she must pivot the finger around the edge of the block to maintain constant pressure into the block and avoid leaving the surface. Detecting the edge is impossible when the finger is treated as a point

and is difficult even when the finger is modeled as an arc segment. Without contact information, one must detect the subtle height drop that occurs as the arc segment approaches the corner. Quite often, subjects overshot the edge of the block under this test condition. In contrast, contact information provides a clear cue that one has reached the edge of the block.

7 Conclusions

This work presents a novel device for displaying contact centroid location along with force feedback during haptic interactions. The data show that contact location information significantly improves contour following capabilities, reducing completion time and failure rate. Subjects also commented that the task was easier to complete and felt more natural with the addition of contact location display. This result complements previous work on curvature discrimination and object motion judged via contact display [9]. Together, these findings indicate that this simple device is a valuable addition to traditional force feedback for virtual and remote exploration and manipulation.

The contact location display system extends the paradigm of standard force-based haptic rendering by providing local tactile information. With this approach, the finger is no longer modeled as a point, but rather as an arc, similar to the commonly used spherical proxy. In contrast to previous force-only haptic interactions, the addition of contact location necessitates anticipation of collision to pre-position the tactile element. A method for predicting nearest contact based on configuration space was described.

The current implementation of the haptic environment was adequate for the contour following used in our experiment. However, while using the system for other types of exploration, we noticed that some contact signals can be misleading because friction is not presently rendered. Users expect the virtual world to match their experience in real interactions, in which friction is ubiquitous. Without simulated friction, differentiating between rolling and sliding behaviors is nearly impossible.

This investigation suggests many future developments. The addition of friction to the environment model will improve the realism of local fingertip exploration. A rotational brake on the tactile cylinder could be used to render the contrast between rolling and sliding contacts, as computed by the friction model. Conversion to a two degree of freedom tactile element would enable display of lateral as well as proximal/distal contact motion. Finally, we believe that the development of a multi-fingered contact location display system would be particularly useful for dexterous manipulation, allowing users to feel object geometry and changes in contact configuration that cannot be conveyed with force feedback alone.

Acknowledgments

This work is supported by the National Science Foundation under grant NSF/IIS-0099636 and under Katherine Kuchenbecker's NSF Graduate Research Fellowship. The authors also thank the reviewers for their helpful comments.

References

- [1] C. Hasser and J. Weisenberger. Preliminary evaluation of a shape-memory-alloy tactile feedback display. In *Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 1993.
- [2] D. Kontarinis, J. Son, W. Peine, and R. Howe. A tactile shape sensing and display system for teleoperated manipulation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 641–646, May 1995.
- [3] S. J. Lederman and R. L. Klatzky. Hand movements: A window into haptic object recognition. *Cognitive Psychology*, 19:342–368, 1987.
- [4] T. Lozano-Perez. Spatial planning: a configuration space approach. *IEEE Transactions on Computers*, C-32(2):108 – 20, February 1983.
- [5] T. H. Massie and J. K. Salisbury. The phantom haptic interface: A device for probing virtual objects. In *Proceedings of the ASME Winter Annual Meeting, Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, November 1994.
- [6] G. Moy, C. Wagner, and R. Fearing. A compliant tactile display for teletaction. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3409–3415, April 2000.
- [7] D. Pawluk, C. V. Buskirk, J. Killebrew, S. Hsiao, and K. Johnson. Control and pattern specification for a high density tactile array. In *Proceedings of the ASME Dyn. Sys. and Control Division, International Mechanical Engineering Congress and Exposition*, pages 97–102, New York, NY, USA, 1998.
- [8] W. R. Provancher. *On Tactile Sensing and Display*. PhD thesis, Department of Mechanical Engineering: Stanford University, 2003.
- [9] W. R. Provancher, K. J. Kuchenbecker, G. Niemeyer, and M. R. Cutkosky. Perception of curvature and object motion via contact location feedback. In *Proceedings of the International Symposium on Robotics Research*, October 2003.
- [10] S. Springer and N. Ferrier. Design and control of a force-reflecting haptic interface for teleoperational grasping. *Journal of Mechanical Design*, 124:277–283, 2002.
- [11] T. Yoshikawa and A. Nagura. A three-dimensional touch/force display system for haptic interface. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 2943–2951, 1999.
- [12] C. Zilles and J. Salisbury. A constraint based god-object method for haptic display. In *Proceedings of the International Conference on Intelligent Robots and Systems*, 1995.