Spanning large workspaces using small haptic devices

François Conti
conti@robotics.stanford.edu

Oussama Khatib
ok@robotics.stanford.edu

Robotics Laboratory
Computer Science Department
Stanford University
Stanford, California 94305, USA

Abstract

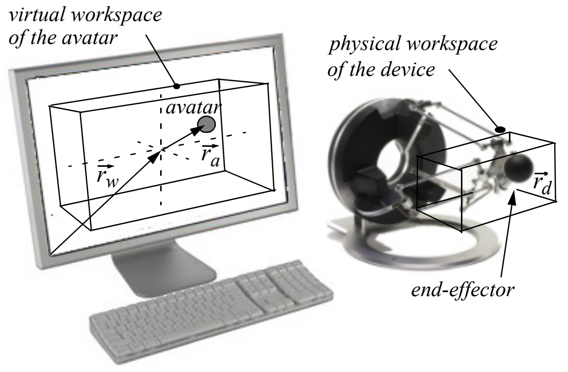
Exploring large virtual environments using a small haptic device with limited-workspace capabilities is a challenging task because the user very quickly reaches the borders of the physical workspace of the device. In the case of a computer mouse, this problem is solved by lifting the device off the table and repositioning it at a different location. With most ground-based haptic devices such indexing procedure is not possible and requires the use of an additional switch to decouple the device from the cursor and allow the user to relocate the end-effector at the center of the physical workspace. Below certain physical workspace dimensions such indexing methods become cumbersome to the operator and therefore different control paradigms are required.

This paper presents a new approach referred to as Workspace Drift Control which progressively relocates the physical workspace of the device mapped inside the virtual environment towards the area of interest of the operator without disturbing his or her perception of the environment. This technique uses the fact that people are greatly influenced by what they perceive visually and often do not notice small deviations of their hand unless that small deviation has a corresponding visual component.

1 Introduction

A haptic device allows a user to interact with a computer via the sense of touch by simulating and rendering contact forces with virtual objects. A typical interface takes the form of a miniature robot arm with sufficient linkages to allow a 2D or 3D workspace. Motors or brakes control the motion of the links which are selectively constrained in order to represent interactions with physical objects. [2, 3, 4, 5, 6, 7, 8, 9, 10]

The user interacts with the device via a gripper or stylus attached at the end-effector of the moveable framework and typically moves a displayed graphical object across a computer screen. Clearly the choice of the displayed object is dependant on what is being simulated and on the devices capabilities. For our purposes we will define an avatar as a virtual representation of the user through which physical interaction with the virtual environment occurs [16]. For instance a surgical tool may be thought of as an avatar when simulating an intervention. A cursor, on the other hand, can be used to represent the position of the mouse being held by the operator. The operator controls the avatar's position inside the virtual environment. When contact takes place between the user's avatar and the virtual objects, action and reaction forces occur. Such forces are regulated by the type of contact supported by the avatar and by its geometry. [11, 12, 13]



$\vec{r}_w$: location of the virtual workspace inside the virtual environment.
$\vec{r}_a$: position of the avatar within the virtual workspace.
$\vec{r}_d$: position of the end-effector in reference to the device base.

Figure 1 - Illustration of a typical 3-dof haptic interface connected to a computer and controlling the position of an avatar inside a virtual simulated environment.

A haptic device is typically used as a position control device in which displacement of the end-effector is directly correlated to displacement of the avatar displayed on the screen. This displacement correlation may not be a one-to-one correspondence, since the avatar position may be scaled according to a constant mapping from the device position. (e.g., the device may be moved a distance of one centimeter which causes the controlled avatar to move five centimeters across the screen.) In general, small displacements of the device are scaled to large motions of the avatar to allow the operator to easily reach targets in all areas of the virtual workspace environment displayed onto the computer screen.

The scaled avatar movement scheme works well for coarse motion, when large distances inside the virtual workspace need to be traversed to bring the avatar from one global area to another. Accuracy of the avatar motion is not critical for coarse motion, but for tasks in which accurate positioning of the avatar is needed, the large scaling of device movement to avatar movement makes a target acquisition task physically impossible for the user.

Ballistic tracking [16] is typically used to alleviate the scaling problem for fine positioning of the controlled object. Ballistics refers to the technique of varying the scaling between the motion of a physical device and the motion of a displayed avatar depending upon the velocity of the device in its workspace. The assumption is that if the user is moving the interface very quickly, the user is likely to be performing a "coarse motion" task inside the virtual environment, and therefore the device controller scales small motions of the interface to large motions of the avatar. Conversely, if the user is moving the device very slowly, the user is likely to be performing a fine positioning task on the screen, and the controller scales small motions of the device to small motions of the avatar.

When using ballistics, moving the device in one direction quickly and then moving it back in the other direction slowly may create a situation where the physical device has returned to its starting position but the avatar is positioned far away from its starting position. This illustrates that the frame of the avatar and the frame of the device have shifted or become offset. If this offset becomes too large, the user may not be able to reach some parts of the virtual workspace within the range of motion of the device.

In a typical, open-workspace interface, the offset is corrected through a process called "indexing." Indexing is achieved in a typical mouse interface [1] by lifting the mouse off the table and repositioning it after the mouse has reached the edge of its available workspace, while the cursor remains fixed in position. However, most force feedback devices are grounded to their base and require the use of an additional input device, such as a user switch for instance, to inform the controller to uncouple the device from the avatar and let the operator reposition the device at the center of its physical workspace. Unfortunately, with limited-workspace devices, indexing becomes cumbersome and highly interferes with the operator since he or she needs to constantly perform the offset correction.

Since ballistics needs indexing to restore the frame offsets, and since ballistics and indexing are both traditional mouse techniques that conflict with typical ground-based haptic devices, a more transparent solution is needed that reconciles both the ballistics and the indexing problem in force feedback interface devices without interfering with the operator.

The rest of this paper is organized as follows: sections 2, 3 and 4 describe position, ballistic and rate control algorithms in more depth and in section 5 we introduce our new control paradigm which automatically centers the physical workspace of the device towards the region of interest of the operator. Section 6 illustrates our initial results and a conclusion is exposed in section 7.

2 Position control

Position control is one of the most common control paradigms used with computer mice or haptic interfaces, and refers to a mapping in which displacement of the device in physical space directly dictates displacement of the avatar in virtual space. The mapping can have an arbitrary scale factor, but the fundamental relation between mouse displacements and graphical object displacements should be present.

In an absolute device-to-world mapping (see figure 1), there is a direct correspondence between the boundaries of the device workspace and the boundaries of the virtual workspace as expressed in equation 1. The increments of the device position are mapped directly to the virtual workspace with a scaling factor k_s; the dimensions of the workspace of the haptic interface can directly correspond to the dimensions of the virtual workspace. If the working volume of the virtual environment is changed, then the scaling factor can be changed to maintain the direct correspondence.

$$\text{(eq.1)} \qquad \vec{r}_a = k_s \cdot \vec{r}_d + \vec{r}_w$$

While this approach allows the operator to navigate through larger virtual environments with smaller haptic interfaces, using a large scale factor reduces the operator's ability to perform fine displacements of the avatar. When force feedback capabilities are engaged, the loss of spatial resolution inside the virtual environment seriously affects haptic performance and stability.

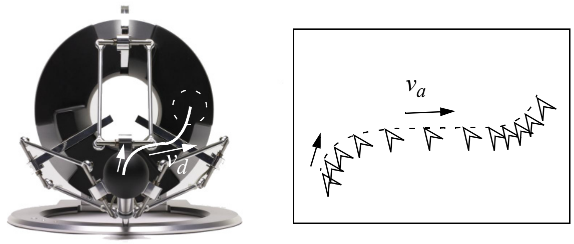
3 Ballistic control

Ballistics addresses the loss of spatial resolution when large scale factors k_s are used by defining a mapping that is dependent on the velocity that the device is currently traveling at in the interface workspace [16]. Ballistics helps to provide accurate control of an avatar when the user wishes to coarsely position the controlled object, (e.g., move a mouse cursors from one object on the screen to another across a large region of the screen). This type of control requires that the avatar be very sensitive to device movements so that the avatar will fly rapidly across the screen. Ballistics also helps to provide accurate control of an avatar when the user wishes to finely position the controlled object, (e.g., to home in on a particular position). This type of control requires the avatar be less sensitive to device movements to allow fine motions. Often, both methods are combined in a single device movement by the operator: first the user swings the avatar quickly into a general region and then he or she homes the avatar in on the target, back-tracking to the target if the avatar overshoots the target.

When the device is moved quickly, ballistics assumes that coarse positioning of the cursor is desired and a large distance is mapped to the avatar. When the device is moved more slowly, ballistics assumes that a finer positioning of the avatar is desired and therefore a smaller distance is mapped for the same motion of the interface.

Equation 2 expresses the relative mapping between the motion of the avatar and the device during a time interval of *dt* where $\overrightarrow{dr_d}$ represents the distance through which the haptic device has been moved and $\overrightarrow{dr_a}$ represents the distance which the avatar covers during that same period of time. $k(v_d)$ is a constant that increases slightly based on the current velocity v_d of the device, as explained above.

$$\overrightarrow{dr_a} = k(v_d) \cdot \overrightarrow{dr_d} \qquad \text{(eq.2)}$$



trajectory of the end-effector *trajectory of the avatar*

Figure 2 - Under ballistic control strategy, faster motions of the device end-effector translate into larger displacements of the avatar.

4 Rate control

Rate control is also a commonly used control strategy that refers to a mapping in which the position of the interface is abstractly mapped to move some kind of mechanism or object. In a computer haptic simulation, where rate control is applied, there is not a direct physical mapping between the physical device motion and the avatar motion. Thus, rate control paradigm is fundamentally different from position control in that the interface can be held steady at a given position but the controlled avatar is in motion at a commanded or given velocity, while the position control paradigm only allows the controlled computer object to be in motion if the interface device is in motion.

For example, a common form of rate control is a velocity derived abstraction in which displacement of the user object dictates a velocity of the computer object, such as a vehicle or other graphical object displayed on the screen. The further the device is moved from the original position, the greater the velocity of the controlled avatar. Such control paradigms are very popular in robotic teleoperation where velocity (or acceleration) of a vehicle is dictated by the displacement of, for example, a joystick. [14]

Equation 3 expresses the displacement $\overrightarrow{dr_a}$ of an avatar under rate control during a period of time *dt*. $\overrightarrow{r_{d0}}$ corresponds to the device origin position at the center of its physical workspace and $\vec{r_d}$ to the current position of the device.

$$\overrightarrow{dr_a} = k_r \cdot \left(\vec{r_d} - \overrightarrow{r_{d0}}\right) \qquad \text{(eq.3)}$$

While rate control strategies allow users to control a simulated object through an infinite workspace without the disadvantages of indexing, such control paradigms also act as low pass filters in position control and remove the ability for the operator to perform fast motions of the avatar in different directions. Thus haptic tasks like scratching a rough surface or tapping against a hard object are no longer possible using such control models.

5 Workspace Drift Control

An essential observation about human perception is that people are greatly influenced by what they perceive visually and often do not notice small deviations of their hand or other physical members of their body in physical space unless that small deviation has a corresponding visual component [18, 19, 20, 23, 24, 25]. This dichotomy between the physical and visual experience has been used previously in haptics to simulate force shading for instance [26] or to render height fields by using only two degrees of freedom planar haptic displays [27].

In our new control strategy, we make use of this observation about human perception to create an imperceptible drift of the physical workspace of the haptic device towards the area of interaction of the avatar. Slowly shifting the workspace of the device when the avatar is in motion instigates the user to unconsciously correct this drift while executing at the same time a task with the device.

So that the workspace drift remains imperceptible to the user several conditions are required: Firstly, a workspace drift may only occur when the user's hand is in motion, otherwise the user would perceive such drift as an unrelated event to the motion of his or her hand. Secondly, the velocity of the workspace drift should remain proportional to the velocity of the user's hand so that the magnitude of the spatial distortion between the physical and visual experience remains below a boundary level which we will discuss later on. Finally a small drift of the workspace should not create any large variations of forces at the end-effector which would be perceived as perturbations by the operator.

In the following figures (3-6), we illustrate our model by using a 2 degree-of-freedom haptic device [4]. The same algorithm is used with a 3 dof haptic device and is presented in section 6.

Figure 3 illustrates the hand of a user holding an end-effector located at the center of the physical workspace of the device. The device directly dictates the position of the avatar as described in equation 1.

To preserve the first two conditions stated above, the velocity of the workspace drift $\overrightarrow{v_{wd}}$ is defined by multiplying the instant velocity of the device by its current offset position from its point of origin. Finally this result is multiplied by a scalar k_d/R where R corresponds to the radius of the smallest sphere that encloses the device workspace and where k_d expresses the drift factor which dictates the maximum level of distortion between the visual and physical representation when the device is located at the edge of its physical workspace (worst case condition).

$$\overrightarrow{v_{wd}} = \frac{k_d}{R} \cdot \left|\overrightarrow{v_d}\right| \cdot \left(\vec{r_d} - \overrightarrow{r_{d0}}\right) \qquad \text{(eq.4)}$$

The velocity of the virtual workspace drift (equation 5) is obtained by multiplying the device workspace drift by the scaling factor k_s. Once the position $\vec{r_d}$ of the workspace updated we finally compute the position of the avatar $\vec{r_a}$ using equation 1.

$$\overrightarrow{v_{wa}} = k_s \cdot \overrightarrow{v_{wd}} \qquad \text{(eq.5)}$$

In figure 4 the operator moves the end-effector towards the edge of the physical workspace of the device; this displacement is directly mapped to the avatar which moves towards the virtual object in the scene.

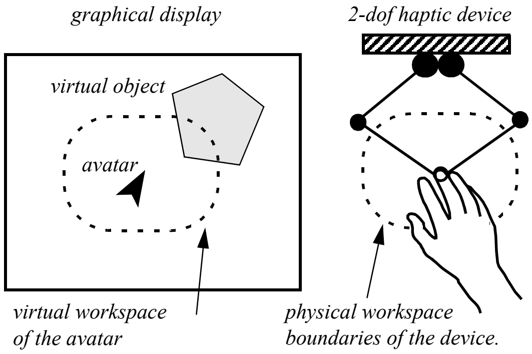


Figure 3 - The hand of the operator is holding the end-effector of the device which is positioned at the center of its physical workspace. The physical workspace is mapped inside the virtual environment (virtual workspace) and partially includes the virtual object.

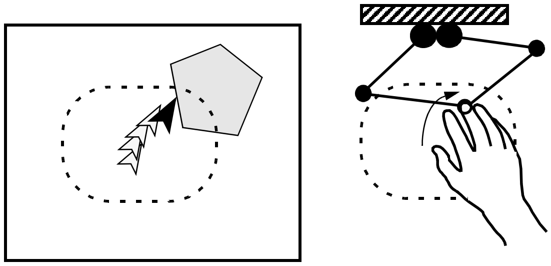


Figure 4 - The operator moves the avatar toward the virtual object. At the same time the device reaches the border of its physical workspace.

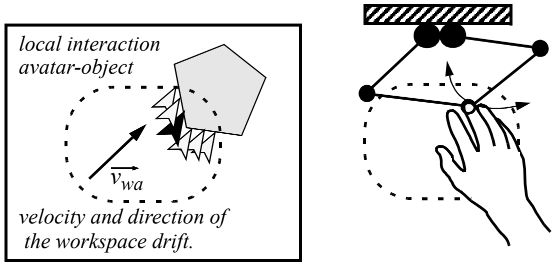


Figure 5 - The operator interacts locally with the virtual object. The virtual workspace drifts towards the avatar.

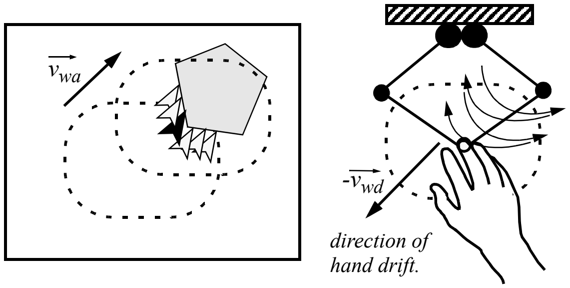


Figure 6 - The workspace drift instigates the operator to move his or her hand towards the center of the workspace of the device.

Finally figures 5 and 6 illustrate the user locally interacting with the virtual object. This task is illustrated by small motions of the avatar in different directions around the edges of the virtual object. During this time the controller progressively moves the workspace of the device towards the avatar, therefore restoring the device at the center of its physical workspace.

When the device is located near its origin the drift is negligible but its amplitude increases noticeably once the user moves towards the edges of the workspace creating an important distortion between the motion of the operator's hand and that of the avatar. The drift factor is expressed by k_d and represents the maximum bend that can occur when the workspace drift is perpendicular to the motion of the end-effector and when the end-effector reaches the edges of the workspace. This situation is illustrated in figure 7.

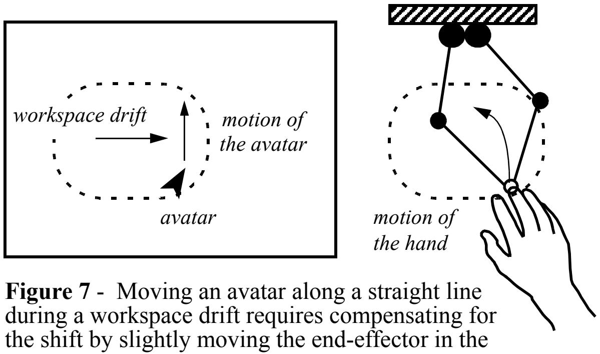


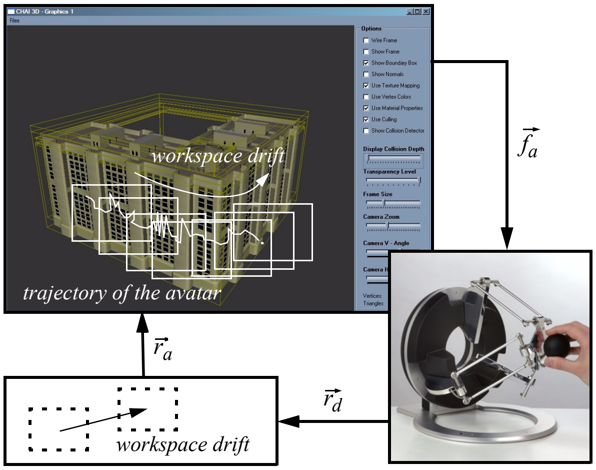
Figure 7 - Moving an avatar along a straight line during a workspace drift requires compensating for the shift by slightly moving the end-effector in the opposite direction of the workspace drift.

6 Results

Our Workspace Drift Controller was directly integrated within the software drivers of an Omega Haptic device, therefore decoupling its implementation from the application. Initial experiments were performed by manipulating large objects inside a 3D Haptic Viewer [15] which integrated a finger-proxy type haptic rendering algorithm. Since the dimensions of the virtual objects were up to 10 times the size of the actual workspace of the avatar (see figure 8), without using the workspace drift control algorithm, the user could only explore a very small portion of the environment (1/10th). When the workspace drift controller was engaged, the user could gradually explore further portions of the scene.

As presented in equation 4, the workspace drift rate is regulated by k_d which directly dictates the maximum level of distortion between the physical and visual representations at the edge of the physical workspace of the device. We provided a slide bar to the user to adjust the drift gain k_d. Initial results showed that for a 3-DOF Omega Haptic Device with a physical workspace radius of R = 0.075m (75mm), a distortion factor of 30% ($k_d = 0.3$) was unnoticed by the operators. Beyond this limit ($k_d > 0.3$) haptic artifacts were observed when the user was interacting with virtual objects; if a workspace drift occurred during this time, the user would experience a slip: the sensation of the object sliding under his or her hand.

To allow the user to cross large regions of the workspace, ballistic control was also integrated into our controller and was only triggered when fast motions of the device occurred. A demonstration program with its source code is available on the CHAI 3D server and can be downloaded at the following address:
http://www.chai3d.org/projects/wdc/index.html



Options
☐ Wire Frame
☐ Show Frame
☑ Show Boundary Box
☐ Show Normals
☑ Use Texture Mapping
☐ Use Vertex Colors
☑ Use Material Properties
☑ Use Culling
☐ Show Collision Detector

Figure 8 - Demonstration within a 3D environment. The virtual workspace is constantly relocated towards the avatar, therefore maintaining the users hand near the center of the physical workspace of the device.

7 Conclusion and future work

The Workspace Drift Control algorithm presented in this paper has proven to significantly improve workspace usage of haptic interfaces without introducing visual or haptic artifacts to the operator when reasonable drift factors are chosen. With this new control approach the operator can explore much larger workspaces without losing spatial resolution through high scaling factors and thus avoid the drawbacks of indexing.

Future work includes extending this methodology to haptic devices with higher numbers of degrees of freedom and pursuing further studies in contact space when multiple collisions and interactions occur between the avatar and the virtual environment.

Additional experiments using different types of haptic devices and a larger number of human subjects will also help us better define the limits of this approach when people begin to perceive haptic and visual artifacts as high spatial distortions are applied to both the physical and visual representations of the virtual scene.

8 References

[1] D. Englebart, "X-Y Position Indicator for a Display System", U.S. Patent 3,541,541, November 1970.

[2] S.C. Jacobsen, F.M Smith, E.K Iversen and D.K Backman, "High Performance, High Dexterity, Force Reflective Teleoperator" in Proc. 38th Conf. Remote Systems Technology, (Washington, D.C.), pp. 180-185. 1990.

[3] T. Massie and J.K. Salisbury. "The phantom haptic interface: A device for probing virtual objects." Proceedings of ASME Haptic Interfaces for Virtual Environment and Teleoperator Systems In Dynamic Systems and Control 1994, volume 1, pages 295--301, Chicago, IL, November 1994.

[4] C. Ramstein and V. Hayward, "The Pantograph: A Large Workspace Haptic Device For A Multi-Modal Human-Computer Interaction" CHI'94, Conference on Human Factors in Computing Systems ACM/SIGCHI Companion-4/94. pp. 57-58. April 1994, Boston, MA.

[5] D. A. Lawrence, L. Y. Pao, M. A. Salada and A. M. Dougherty, "Quantitative Experimental Analysis of Transparency and Stability in Haptic Interfaces" Proc. ASME Dynamic Systems and Control Division, (Atlanta, GA), DSC-Vol. 58, pp. 441-449. 1996.

[6] P. J. Berkelman, Z. J. Butler, and R. L. Hollis, "Design of a Hemispherical Magnetic Levitation Haptic Interface Device," 1996 ASME IMECE, (Atlanta, GA), DSC-Vol. 58, pp. 483 488.

[7] V. Hayward, V. Gregorio, P. Astley, O. Greenish, S. Doyon, M. Lessard, L. McDougall, J.Sinclair, I. Boelen, S. Chen, X. Demers, J.-P. Poulin, J. Benguigui, I. Almey, N. Makuc and B. Zhang, X. 1998. "Freedom-7: A High Fidelity Seven Axis Haptic Device With Application To Surgical Training." In Experimental Robotics V, Casals, A., de Almeida, A. T. (eds.), Lecture Notes in Control and Information Science 232, pp. 445-456.

[8] S. Walairacht, Y. Koike and M. Sato:"A New Haptic Display for Both-Hands-Operation:SPIDAR-8"; ISPACS'99(1999 IEEE International Symposium on Intelligent Siganal Processing and Communication Systems, Phuket, Thailand, 569-572(1999.12)

[9] A. Cohen and E. Chen., "Six degree-of-freedom haptic system for desktop virtual prototyping applications." Proceedings of the ASME Winter Annual Meeting, Dynamics Systems and Control, DSC-Vol67}, pages 401--402, Nashville, Tennessee, November 1999.

[10] S. Grange, F. Conti, P. Helmer, P. Rouiller. and C. Baur, "Overview of the Delta Haptic Device" Eurohaptics '01, Birmingham, England, July 2001.

[11] J.E. Colgate, P.Grafing, M.C. Stanley and G.. Schenkel, "Implementation of stiff virtual walls in force-reflecting interfaces". In Proceedings of the IEEE Virtual Reality Annual International Symposium (pp. 202-208). Piscataway, NJ: IEEE Service Center. (1993).

[12] K. Salisbury, D. Brock, T. Massie, N. Swarup, and C. Zilles, "Haptic rendering: Programming touch interaction with virtual objects.". In Proceedings of the 1995 Symposium on Interactive 3D Graphics (pp. 123-130).

[13] D. Ruspini, K. Kolarov and O. Khatib, "The Haptic Display of Complex Graphical Environments", Computer Graphics SIGGRAPH 97 Conference Proceedings, pages 345--352, 1997, ACM SIGGRAPH.

[14] S. Grange, F. Conti, P. Helmer, P. Rouiller, and C. Baur, "The Delta Haptic Device as a Nanomanipulator", SPIE Microrobotics and Microassembly III, Boston, MA, November 2001.

[15] F. Conti, F. Barbagli, K. Salisbury, 2003, "The CHAI Libraries, Eurohaptics", June 2003, Dublin, Ireland.

[16] K Salisbury, F Conti and F Barbagli, "Haptic Rendering: Introductory Concepts", IEEE CG&A, March/April 2004.

[17] J. Mallett, D. Chang, L. Rosenberg, A. Braun, K. Martin; J. Beamer. U.S. Patent 6,292,174

[18] M. J. Morgan. "Molyneux's Question: Vision, Touch and the Philosophy of Perception." Cambridge University Press, New York, US, 1977.

[19] G.A. Gescheider. "Psychophysics Method, Theory, and Application.," Lawrence Erlbaum Associates, New Jersey, US, 1985.

[20] C. Bonnet. "Manuel pratique de psychophysique", Armand Colin, Paris, France, 1986.

[21] S. Zai. "Investigation of Feel for 6dof Inputs: Isometric and Elastic Rate Control for Manipulation in 3D Environments." In Proc. of the Human Factors and Ergonomics Society 37th Annual Meeting, Oct. 1993.

[22] R. R. Ellis and S. J. Lederman. "The Role of Haptic versus Visual Volume Cues in the Size-Weight Illusion." Perception and Psychophysics, 53(3):315-324, 1993.

[23] M. A. Srinivasan, G. L. Beauregard, and D. L. Brock. "The Impact of Visual Information on Haptic Perception of Stiffness in Virtual Environments", In Proc. of ASME Winter Annual Meeting, 1996.

[24] E. B. Goldstein. Sensation and Perception. Brooks/Cole, US, 1999.

[25] A L´Ecuyer, S. Coquillart, A. Kheddar, P. Richard, P. Coiffet "Pseudo-Haptic Feedback: Can Isometric Input Devices Simulate Force Feedback?", IEEE Int. Conf. on Virtual Reality, pages 83-90, New Brunswick, US, 2000.

[26] H. B. Morgenbesser, "Force Shading for Haptic Shape Perception in Haptic Virtual Environments." M.Eng. thesis, MIT, September 1995.

[27] M. D. R. Minsky, , "Computational Haptics: The Sandpaper System for Synthesizing Texture for a Force-Feedback Display." PhD thesis, MIT, June 1995.