Tailor: Creating Custom User Interfaces Based on Gesture

Randy Pausch & Ronald D. Williams

University of Virginia
Thornton Hall
Charlottesville, VA 22903
(pausch@Virginia.edu)
804-982-2211

Abstract

Physical controls for most devices are either "one size fits all" or require custom hardware for each user. Cost often prohibits custom design, and each user must adapt to the standard device interface, typically with a loss of precision and efficiency. When user abilities vary widely, such as in the disabled community, devices often become unusable. Our goal is to create a system that will track user gestures and interpret them as control signals for devices. Existing *gesture recognition* research converts continuous body motion into discrete symbols. Our approach is to map continuous motions into a set of analog device control signals. Our system will allow us to quickly tailor a device interface to each user's best physical range of motion. Our first application domain is a speech synthesizer for disabled users. We expect two major areas of applicability for non-disabled users: in *telemanipulator* interfaces, and as a design tool for creating biomechanically efficient interfaces.

Introduction

The Augmentative Communications Group at the University of Virginia consists of researchers from the Computer Science and Electrical Engineering departments, and from the Medical and Education schools. Our current effort is to create a speech synthesizer for individuals with cerebral palsy, a disability affecting approximately 700,000 Americans [1]. A significant portion of the cerebral palsy population is communicative but non-verbal – although the desire to communicate is present, speech is prohibited by damage to the part of the brain that controls the vocal tract. Most of these individuals do not have enough coordination for handwriting or typing. Although primitive electronic communication aids exist, most are variations on *picture boards,* where the user points or looks at a two-dimensional array of pictures to convey a thought such as "hungry" or "tired."

The Augmentative Communications Group has developed a speech synthesizer based on two-dimensional analog input. The creation of speech involves the coordination of a large number of muscles in the vocal tract. The synthesizer approximates this by receiving the position of the base and tip of the tongue as input signals and then synthesizes the sound produced by that position of the tongue. We have implemented a prototype which synthesizes monotone speech from two analogue signals.

The original research strategy was to attempt to design and build custom input devices for each user of the system. Biomedical engineers constructed various one-dimensional potentiometers to be used in pairs to provide the analog inputs needed to drive the synthesizer. Building custom hardware interfaces is expensive, and cerebral palsy victims often have reduced strength, making control of any physical device cumbersome. As users fatigue, their efficiency with a particular device decreases and several different devices may be needed to accommodate various stages of fatigue.

Our new approach is to create an individual gesture interface for each user. Our software maps body motions, reported by magnetic trackers, into continuous control signals for the speech synthesizer. The only physical effort by the user is to move a part of his body. This software tailoring allows us to create interfaces based on each user's individual abilities, and makes it possible for those interfaces to adapt as the user fatigues.

The idea of user tailoring, or customization, has long been understood as a crucial element in the design of traditional computer interfaces. Text editors allow users to rebind keyboards so that commonly used operations are easier to reach [2]. Mice often allow for alterations in the ratio of device motion to cursor motion, to accommodate variations in user coordination. Some hardware interfaces allow minor customization, such as tilt-lock steering wheels in automobiles.

Customization is necessary when there is a high degree of user idiosyncrasy, *relative to the dexterity required for the task.* While the need for individual tailoring is most apparent in the disabled community, able-bodied users exhibit high idiosyncrasy with respect to tasks where extreme dexterity is required, such as telemanipulation in microsurgery. We expect our techniques to be useful for able-bodied users when they must use interfaces that require high dexterity to complete tasks.

Existing gesture research is dominated by a desire to understand or *interpret* gestures, and is commonly referred to as *gesture recognition.* Our approach is to map continuous data from one or more sensors to a set of continuous device control signals, rather than transforming gestures into symbols. Our primary goal is to create custom gesture mappings for device control. Our secondary goal is to make our mappings dynamically adjust for fatigue and changes in user ability.

A Joystick-Driven Speech Synthesizer

Because our synthesizer is such an unusual device, we first describe how it is able to synthesize speech from two analog inputs. Our articulator driven speech synthesizer produces sounds using the positions and motions of implied articulators in a simulated vocal tract. This form of speech synthesis has been discussed previously in the literature [3-5]. The problem addressed here differs from previous work because we limit the number of articulator control parameters to those that can be provided by a human user in real-time.

Articulator driven synthesis is unnecessary and constraining in the text-to-speech environment, but this approach is directly analogous to the mechanisms of speech production used for normal human conversational speech. A brief review of human physical speech production will be helpful in understanding the articulator driven synthesis approach.

The physical process of speech production can be divided into three parts. First, air is forced through the vocal cords to produce either a voiced or unvoiced glottal excitation. Next, air flow is modified by a series of structures that constitute the vocal tract. Finally, the modified flow is radiated through the lips and nostrils. [6]. The articulators used to produce speech are shown in Figure 1.

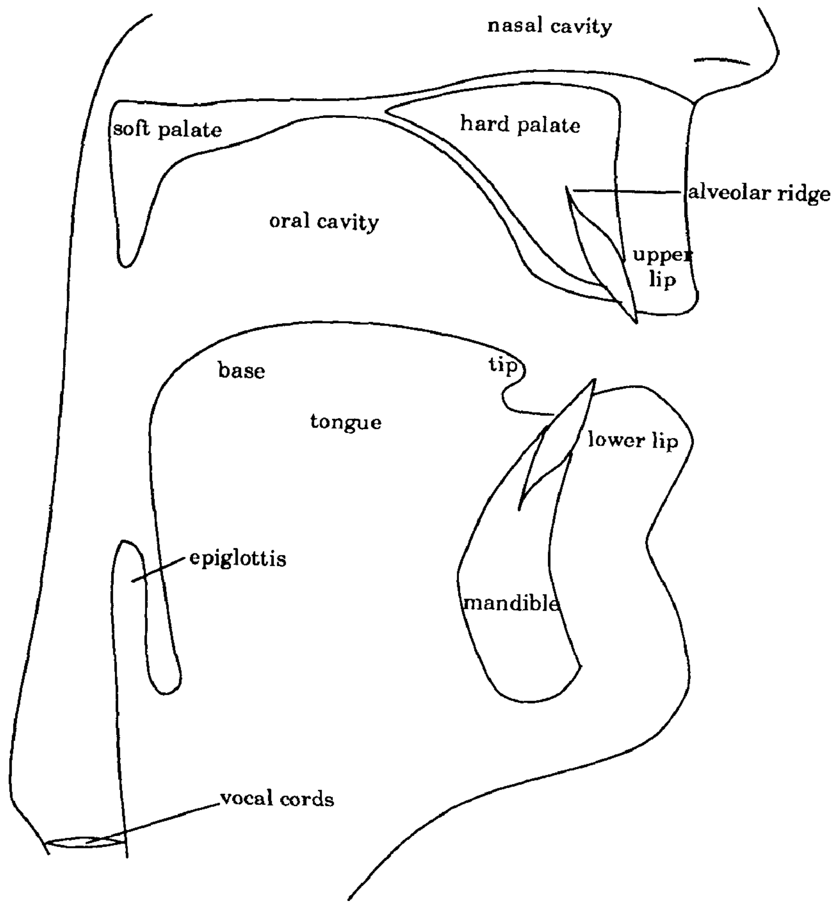

Figure 1: Articulators That Contribute to Sound in the Human Vocal Tract

To change a sound, the articulators are moved from one position to another in continuous motions which give speech its continuous, fluid quality. The tongue is the most important articulator. The jaw, lips, and velum are less important for shaping the speech spectrum [7].

The constraints of the application suggest that we base our system on a physical model of speech that includes glottal excitation and vocal tract adjustments caused by the articulators. Some simplifying assumptions are made to achieve a minimal system. A minimal system is required because the synthesizer must be driven in real-time by a human user through a limited interface. The complexity of this interface must be limited because of the speed with which the user must operate the interface.

Our current model focuses only on the tongue. Since the tongue acts as a continuous modifier of speech sounds, its motion can be modeled as a set of analog signals. These signals represent the control of specific muscles in the vocal tract that move the tongue to its proper configuration for a specified sound. Physically, the tongue can be simplified to a movement of its tip and base [7]. This tip and base movement can be viewed as an orthogonal two-dimensional signal where motion along one axis represents the tip, and motion along the

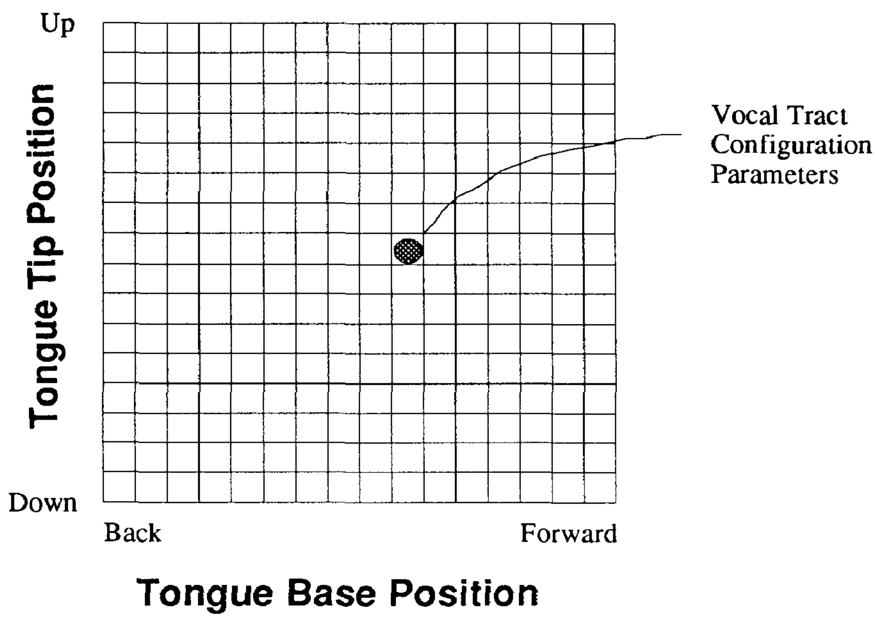

Figure 2: Two-dimensional Tongue Position Grid

other axis represents the base. In this way, the tongue tip and base can be described independently by holding one dimension constant, or together by varying the position along both axes.

The tongue's base and tip position can be mapped onto a two-dimensional grid as seen in Figure 2. As the tongue is moved from one position to another, the grid location is used to calculate the coefficients for use by the speech synthesizer.

This technique, combined with an interpolation scheme, overcomes the transition problem that all discrete-unit synthesizers must address. As an example, consider synthesis based upon the generation of discrete phonetic units for discrete periods of time. The word *same* might be synthesized by concatenating the /*s*/, /!EY!/, and /*m*/ units. Between these units, transients can occur that make the speech sound unnatural [8]. The simplified articulator driven system requires a time trajectory between any two sounds. This trajectory will have synthesis data along its path, so the transitions are continuous, with interpolation being used to smooth these transitions.

The current implementation can be used to produce crude, monotone speech by using a joystick to navigate the tongue position grid [9]. The joystick is a temporary testing device, as the target population does not have the dexterity to control a standard joystick. Early attempts to build interfaces for the synthesizer focused on building analog input devices, such as levers to be placed against the cheek or arm. A number of novel analog devices were devised, including air sacs to detect force, and throat microphones that detect a level of low throat "growl" that some subjects could produce. The intention was to combine two such one-dimensional input devices to provide the analog signals needed for the grid. The difficulty of building effective hardware interfaces, combined with the effects of user fatigue, created major difficulties with this approach.

Tracking Body Motions

Our new strategy is to track three-dimensional user motions and to create custom projections to the two-dimensional format grid for each user. The most obvious advantage of this approach is that we can tailor the interface to each individual's best range of physical motion. Another advantage is that no strength is required to move a physical switch. For the cerebral palsy community, another advantage is that less coordination is required; with a physical interface, the user much first contact the device, and then move it in some way. The final advantage is that a software interface based on motion tracking can be adapted over time to account for development and/or fatigue.

One alternative to tracking body motion is to track eye motion [10-15]. Eye-tracking is not appropriate for our application for several reasons. First, many disabled individuals have trouble controlling their eye movements. Second, using eye-tracking for the speech synthesizer makes it impossible to maintain eye contact or receive visual stimulation while speaking. Third, many disabled users are poor candidates for eye-tracking because they tend to move their heads.

The major disadvantage of passive tracking technologies is the loss of tactile and force feedback. Providing synthetic force and tactile feedback is an issue currently being investigated by researchers at the University of North Carolina at Chapel Hill and the MIT Media Lab [16, 17]. Although force feedback will probably never be viable in a free-motion tracking environment, the creation of lightweight clothing that provides some form of tactile feedback is certainly possible. Our work will concentrate on a model where feedback is provided visually and audibly by the device being controlled, not by the sensing technology.

Gesture recognition has a long history in many contexts, but most research has focused on converting continuous body motion into discrete tokens. Two-dimensional gesture research has included recognizing both printed [18, 19] and cursive [20, 21] handwriting, proofreader's symbols [22], shorthand [23], and interaction techniques such as circling and dragging objects in a direct-manipulation interface [24, 25]. Recognition of three-dimensional gestures has also been attempted, but again the main emphasis has been on converting the body motions into discrete symbols that are interpreted as commands to the system [26, 27]. Systems have attempted to recognize static gestures for the deaf alphabet and motions for a subset of American Sign Language [28, 29]. All of these approaches are based on converting three-dimensional signals into a discrete stream of application commands.

Existing work on mapping gesture into continuous control signals is extremely application dependent. For example, advanced military systems exist that map pilot head motion into weapon trajectories. The pilot's faceshield contains targeting crosshairs, and as the pilot's helmet moves rigidly with his head, the system computes the angle of his gaze [30]. More detailed tracking is performed in three dimensional drawing or sculpting applications [31], and *virtual reality* systems [32-35], where various three-dimensional signals [36-38] are mapped into motions in synthetic worlds shown on traditional or head-mounted displays [32, 39]. These systems perform mappings from position and orientation information, but the mappings are significantly less complicated than those we propose. The notable exception is the recent effort at VPL Research, Inc. to map user motions into synthetic creature motions [40]. sp 2

The Tailoring System

Our experimental setup is shown in Figure 3. One or two magnetic trackers are attached to the subject, at locations determined by the therapist. If only one tracker is used, the mapping problem reduces to mapping a six-dimensional signal (x, y, z, azimuth, elevation, roll) into a two-dimensional signal. There are two possible ways that two trackers will be used. In the first case, they both generate independent data, and the problem becomes a mapping from twelve dimensions to two dimensions. A second use of dual trackers is to use one as a base for the other. For example, if we are measuring head motion relative to the neck, and the subject tends to rock or raise his torso, we may attach the second tracker to the neck and use it as a base to compute the relative motion of the first tracker.

The signals from the trackers are sent via a high-speed serial connection to the mapping CPU. This station displays mapping visualization and interactive controls for the therapist performing the tailoring. The mapping CPU produces one or two continuous signals that are sent to the application CPU. The application CPU is responsible for providing the visual and/or auditory feedback that will guide the user's actions while using the gesture interface. Because the speech synthesizer is a complicated interface to master, we are currently developing simpler one and two dimensional graphical applications that the disabled users will control initially.

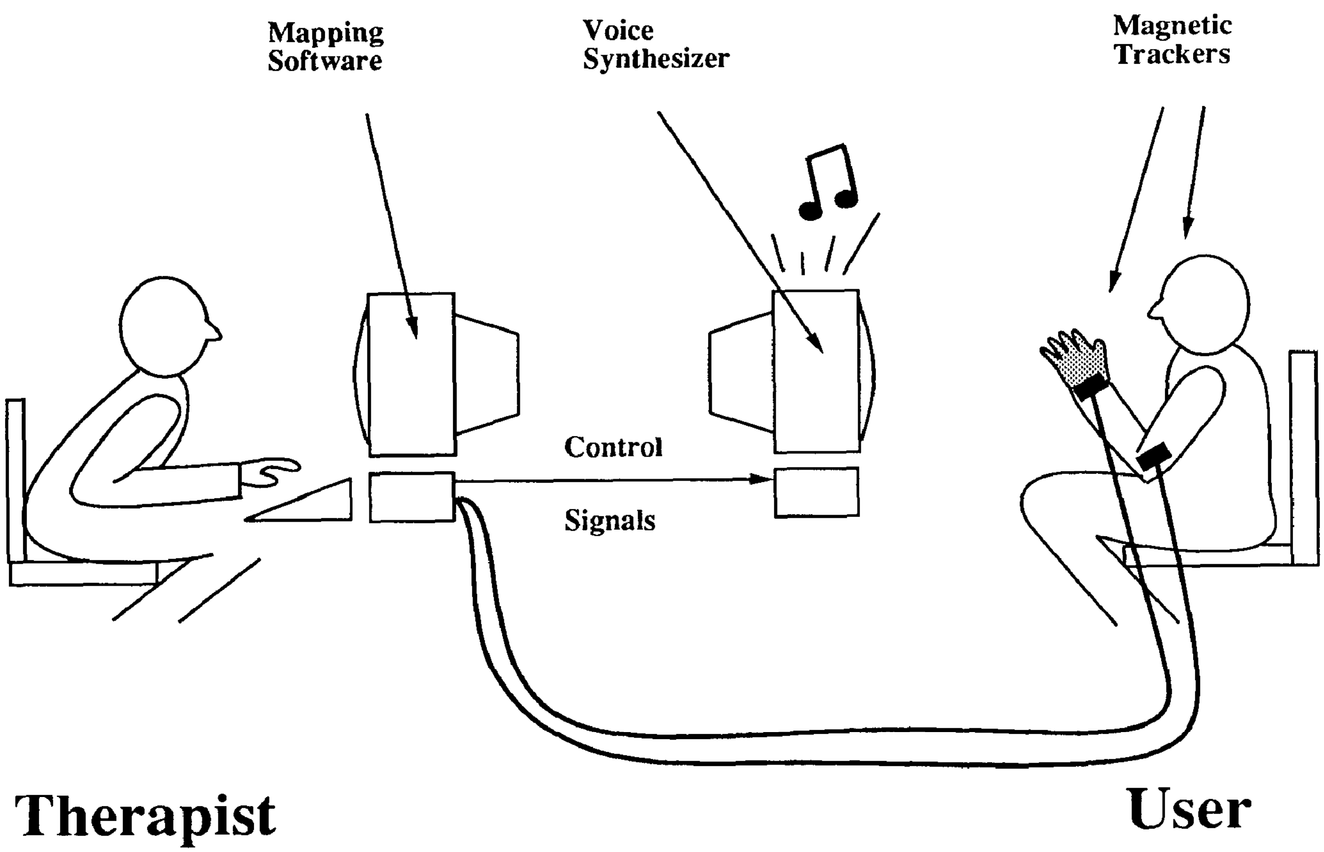

Figure 3: The Experimental Setup

Mapping consists of two basic phases. The *collection phase* determines the comfortable and preferred motions for the user. The *control phase* performs real-time mapping of user motion based on a mapping function created from the data obtained during the collection phase. Although the mappings we create are biomechanically comfortable for each user, there is no reason to expect that they will be easily teachable by the therapist. As the candidate mapping is being used, users notice the results of their motions and experiment to discover the nature of the mapping, rather than having it taught to them by the therapist. Although they will make motions with the intent of changing the device's state, we want them thinking about the device state, not how to make their motions be properly mapped. We note that this may not be immediately apparent from the specific examples used, however these examples were chosen because their mappings are easily displayed geometrically.

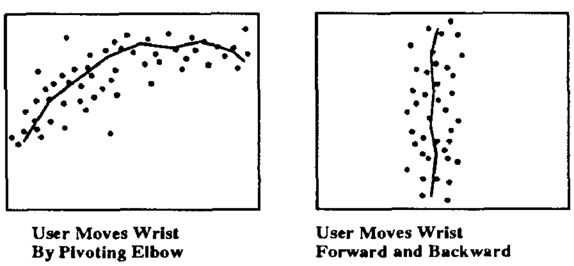

Figure 4: Target Curves

Our current mapping approaches are based on *target curves* and *target surfaces*. We first describe a simple mapping for our current implementation, which provides users with the ability to control a device requiring one continuous input parameter. In this example, the "device" is a video cursor that moves in a horizontal line. During the collection phase, the user is instructed to move the tracker in any manner that is comfortable, while we collect position data from the sensors. During this time, the user receives no visual or auditory feedback from the system. After roughly thirty to sixty seconds, the data is analyzed to determine a curve in three space through which the user would be able to comfortably navigate the tracker.

In order to facilitate the visualization of static diagrams in this proposal, assume that the user had a tracker attached to a wrist, and was told to keep his hand on a horizontal table during the measurement. This effectively constrains his motion to two

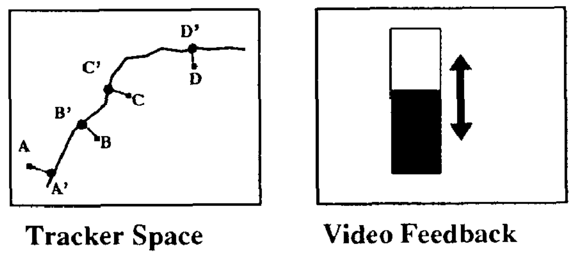

Figure 5: Tracker Space to Device Space

dimensions. Based on the on-screen display of this raw data, the therapist creates a piecewise linear curve though the data, corresponding to an dominant path of motion made by the user during the control phase. This is done by invoking a heuristic, manually specifying the curve, or a combination of both. Figure 4 shows two typical target curves and the data used to form them. The first user pivoted his wrist around his elbow, and the second moved his wrist forward and backward.

During the control phase the user moves the tracker along the target curve and we generate a linear control signal. One end of the curve indicates 0 percent of this signal and the other end indicates 100 percent. Intermediate positions along the target curve indicate intermediate signal values and the signal generates video feedback. The user is not expected to move the tracker precisely along the curve; we map tracker data to the nearest point on the target curve, as shown in Figure 5.

The user never sees the display of the target curves, although the therapist may attempt to explain the mapping to the user. Because the target curve is composed of comfortable motions, our anticipation is that the therapist will often be able to let the user discover the mapping himself. While observing the user actions during the control phase, the therapist may dynamically alter the target curve using his interactive display. The mapping software may also dynamically display alternative target curves created from continuing to observe the user's motions. The therapist may also specify a non-linear mapping from position along the target curve to the values of the device signal. By entering explicit scaling points onto the two graphs, the therapist may adjust the distance along various parts of the curve that the user must move to cause a unit of motion in the device space, as shown in Figure 6.

Although the previous example hypothesized limiting the user's motions to a table surface, target curves reside in three space. The tailoring tools display the raw tracker data as a green, three-dimensional point cloud, with a red target curve running through the cloud. Dynamic markers show the tracker positions in real time during the control phase. The therapist can dynamically rotate his view of the target curve and tracker positions, and dynamically adjust the target curve as the system runs.

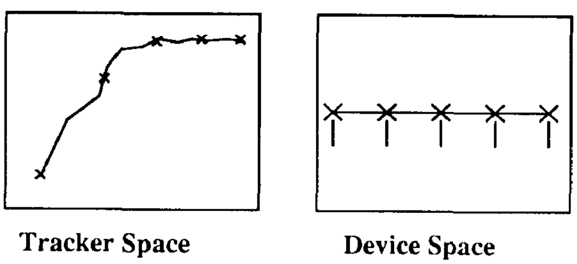

Figure 6: Non-Linear Mappings

Future Research

For some users, it may be possible to use two trackers and two independent target curves to create the two signals needed for the synthesizer. We expect a more common technique will involve the creation of a piecewise planar *target surface.* During the control phase, each user point is mapped to the closest point on the target surface, as shown in Figure 7.

The target surface is decomposed into a grid of planer sections that is then mapped into the grid for the two-dimensional device signal, as shown in Figure 8. The therapist can once again specify a non-linear mapping by stretching the planar patches to alter the transformation to the device signal. As with target curves, we view target surface creation as a joint task between the therapist and the tailoring software.

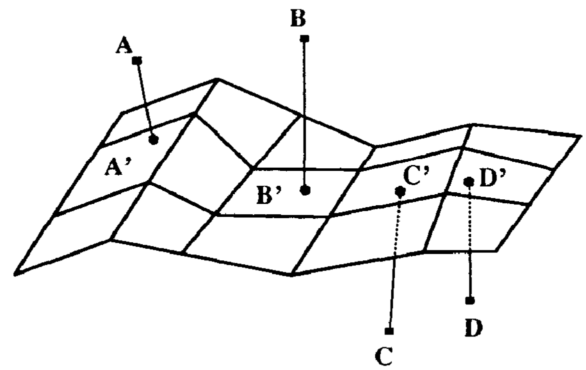

Figure 7: Mapping to a Target Surface

In experiments we have run, humans are very adept at immediately sketching appropriate target curves for two-dimensional data. In three dimensions, it becomes more labor intensive to produce the target curve. We have developed greedy heuristics that start at the densest portion of the cloud and produce basically acceptable curves that a therapist may easily alter.

For clouds where the heuristic's response is not good enough, we have implemented several genetic algorithms [41], that operate by keeping a population of potential solutions and perform geometric "mating" of them in an attempt to produce better "offspring." Both the heuristic and genetic algorithms measure the success of their solutions by a weighted function with two components. The first component is the sum of the distance from each point in the cloud to the nearest point on the curve. The second component is the smoothness of the curve, measured as the sum of the *differences* of the angles between each pair of piecewise connections.

The previous examples all showed mappings based on the positional information from the sensors. We expect some of our mappings to be less geometrically obvious. We will initially concentrate on target curves and surfaces that can visualized by the therapist. Later target curves and surfaces will be in spaces not easily visualized; in those cases, we will create the mapping entirely in software.

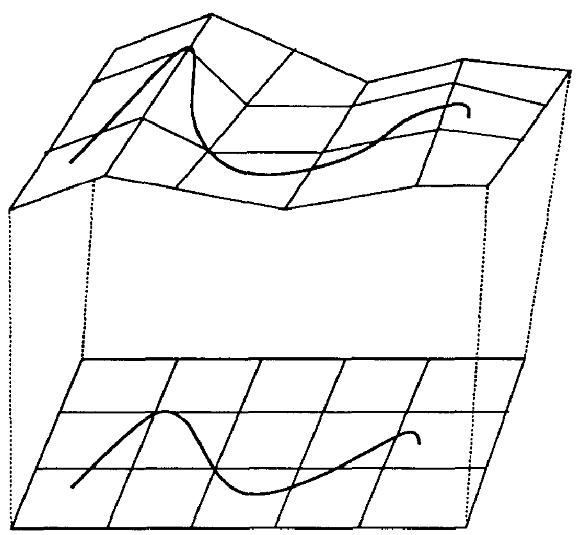

Figure 8: From Surface To Two Signals

Often the tracker motion is not best interpreted in an absolute coordinate system. For example, some sample subjects have had good control of head motion relative to their torso, but tend to stand up or rock their bodies while concentrating on a task. In these cases, tracking head motion alone would be useless. In cases such as these, we will attach one tracker to the torso and treat it as a moving base. The second tracker data will be interpreted relative to the first, and the therapist's display will present clouds and target curves and surfaces as if the user's torso were stationary. We expect that many of our mappings will occur in these anatomically based coordinate systems. We do not expect to create a complex software model of biomechanical motion, which would be beyond the scope of our efforts.

Although we can not completely predict the advanced mappings we will construct, we can hypothesize several mapping strategies that may be useful. For some users, it may be more appropriate to examine derivative rather than positional information. Another aspect we anticipate with advanced mappings will be the scaling of time as the control signals are sent to the application. Many disabled users have reduced speed of motion, and it may be appropriate to detect motion over a period of time and then time compress the signals being sent to the application. To keep the mapping and application synchronized, during some time intervals, no signal will be sent to the application. This is appropriate for the speech synthesizer, where we would encapsulate a spoken phrase at slower than real-time, and then compress it before sending it to the synthesizer.

Our approach creates comfortable mappings for each user, but the targets are somewhat abstract. We are currently experimenting with physical *guides* to focus the user's motions. Our standard example is to instruct the user to run his hand over a teddy bear whose stomach has been specified as the target surface. In this way, we can quickly turn any existing physical object into a input device. The limiting factor is that the user must have a comfortable range of motion over the surface of the object.

In order to adapt for fatigue over time, our initial plan is to continue to add all user tracking points to the cumulative cloud as the user controls the device. As the cloud shifts, we will make our heuristics and genetic algorithms adjust the mapping in real time. The genetic algorithms are more appropriate for this task than the heuristic, which runs in a batch mode to determine a single solution. In situations where fatigue becomes a dominant concern, the therapist may choose to use a different attachment point, or elect to purge the current cloud and begin with only the new motions in order to speed up adaptation.

Summary

Our initial results have shown that children with cerebral palsy have been able to successfully control one dimensional signals using our system. Our users have successfully controlled devices through the use of one dimensional target curves in three-space. The interface adapts to each user's physical capabilities, and does not require the users to exert force on physical controls. These factors have produced significant qualitative results regarding the relaxation and attention span of our users.

Although the initial motivation for this work is building a speech synthesizer interface, our research has obvious application for allowing disabled individuals to control other devices. For example, if used in combination with an on/off switch, a target surface would allow users to control applications by mimicing a mouse. We also expect two major areas of applicability for non-disabled users: in telemanipulator interfaces, and as a design tool for creating biomechanically efficient interfaces.

Telemanipulators allow users to manipulate physical devices in one location, while robotic actuators perform the intent of that control at a second location. In some cases, the motivation is inaccessibility, such as operation in a hazardous or distant site. Other motivations involve issues of scale; certain forms of microsurgery involve manipulating actuators in extremely small spaces. One way to gain accuracy is by altering the scale of the controls to the actuators. When accuracy is increased, speed is decreased because now the control motions are larger. Since each user has a different level of manual dexterity, having the ability to dynamically scale and adjust the controls on such systems would be advantageous.

A second application of our techniques for non-disabled users would be in the design of biomechanically interfaces, either to be used with tracking technology, or as a guide in creating efficient hardware interfaces. Our system will allow designers to instrument and prototype control motions, either for a single individual or across a range of users. User control design is currently in many respects a trial and error process, where hardware prototypes are built for each iteration. Building hardware prototypes is prohibitively expensive for many applications. Although the computer has gained acceptance as a design tool for many products, user control design is still primarily done with physical models, and is often the weakest part of the design of a product.

Our next efforts will include target surfaces which allow users to produce two simultaneous control signals. Our longer term goals are to dynamically adjust for user fatigue, and to attempt to move to a three dimensional model for the speech synthesizer.

References

1. *What Everyone Should Know About Cerebral Palsy*, United Cerebral Palsy Association of Westchester County, Inc., 1985.
2. R. Stallman, *GNU Emacs Manual*, Free Software Foundation, October 1986.
3. W. L. Henke, Preliminaries to Speech Synthesis Based upon an Articulatory Model, *Proceedings of the IEEE Conference on Speech Communication and Processing*, New York, 1967, 170-182.
4. C. H. Coker, Synthesis by Rule from Articulatory Parameters, in *Speech Synthesis*, J. L. Flanagan and L. R. Rabiner (editors), Dowden, Hutchinson & Ross, Inc., Stroudsburg, PA, 1973, 396-399.
5. M. Haggard, Experience and Perspectives in Articulatory Synthesis, *Frontiers of Speech Communication Research*, London, UK, 1979, 259-274.
6. L. R. Rabiner and R. W. Schafer, *Digital Processing of Speech Signals*, Prentice Hall, Englewood Cliffs, NJ, 1978.
7. W. R. Zemlin, *Speech and Hearing Science*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1968.

8. D. G. Childers, K. Wu and D. M. Hicks, Factors in Voice Quality: Acoustic Features Related to Gender, *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, New York, 1987, 293-296.
9. A. Girson and R. Williams, Articulator-Based Synthesis For Conversational Speech, *International Conference on Acoustics, Speech, and Signal Processing*, April, 1990.
10. L. R. Young and D. Sheena, Survey of Eye Movement Recording Methods, *Behavior Research Methods and Instrumentation* 7(1975), 397-429.
11. J. Merchant, R. Morrissette and J. L. Porterfield, Remote Measurement of Eye Direction Allowing Subject Motion Over One Cube Foot of Space, *IEEE Transactional of Biomedical Engineering 21*(1974), 309-317.
12. R. A. Bolt, Gaze-Orchestrated Dynamic Windows, *Computer Graphics 14*,3 (August 1981), 109-119.
13. R. A. Bolt, Eyes at the Interface, *Proceedings of the ACM SIGCHI Human Factors in Computer Systems Conference*, 1982, 360-362.
14. C. Ware and H. T. Mikaelian, An Evaluation of an Eye Trackers as a Device for Computer Input, *Proceedings of the ACM SIGCHI Human Factors in Computer Systems Conference*, 1987, 183-188.
15. R. J. K. Jacob, Eye-tracking, *Course Notes, ACM SIGGRAPH*, 1989.
16. M. Ouh-young, M. Pique, J. Hughes, N. Srinivasan and F. P. B. Jr., Using a Manipulator For Force Display in Molecular Docking, *IEEE Robotics and Automation Conference 3*(April, 1988), 1824-1829.
17. M. Minsky, Virtual Environments and Interactity: Windows to the Future, *ACM SIGGRAPH Panel*, July 1989.
18. C. Suen, M. Berthold and S. Mori, Automatic Recognition of Hand-Printed Characters - The State of the Art, *Proceedings of IEEE 68*,4 , 469-487.
19. R. Hayes, True Notebook Computing Arrives: The GRiDPad, *Byte*, December 1989, 94-95.
20. R. W. Ehrich and K. J. Koehler, Experiments in the Contextual Recognition of Cursive Script, *IEEE Transaction on Computers 24*,2 , 182-194.
21. C. C. Tappert, Cursive Text Recognition by Elastic Matching, *IBM Journal of Research and Development 26*,6 , 765-771.
22. M. L. Coleman, Text Editing on a Graphic Display Device Using Hand-Drawn Proofreader's Symbols, in *Pertinent Concepts in Computer Graphics*, M. Faiman and J. Nievergelt (editors), University of Illinois Press, 282-290.
23. R. Dye, A. F. Newell and J. L. Arnott, An Adaptive Editor for Shorthand Transcription Systems, *Human-Computer Interaction-Interact '84*, , 157-161.
24. J. C. Jackson and R. J. Roske-Hofstrand, Circling: A Method of Mouse-Based Selection Without Button Presses, *Proceedings of the ACM SIGCHI Human Factors in Computer Systems Conference*, May, 1989, 161-166.
25. M. Minsky, Manipulating Simulated Objects with Real-world Gestures using a Force and Position Sensitive Screen, *Computer Graphics 18*,3 , 195-203.
26. W. Buxton, E. Fiume, R. Hill, A. Lee and C. Woo, Continuous hand-gesture driven input, *Proceedings of Graphics Interface '83*, , 191-195.
27. R. Bolt, Put-That-There: Voice & Gesture at the Graphics Interface, *Computer Graphics, 14*,3 (1980), 262-270.
28. J. Dramer, The Talking Glove in Action, *Communications of the ACM 32*,4 (April, 1989), 515.
29. J. Loomis, H. Poizner, U. Bellugi, A. Blakemore and J. Hollerbach, Computer Graphic Modeling of American Sign Language, *Computer Graphics 17*,3 (July 1983).
30. T. A. Furness, *Super Cockpit: Virtual Crew Systems*, Armstrong Aerospace Medical Research Laboratory, 1988.
31. C. Schmandt, Spatial Input/Display Correspondence in a Stereoscopic Computer Graphic Work Station, *Computer Graphics 17*,3 (July, 1983), 253-261.

32. S. S. Fisher, M. McGreevy, J. Humphries and W. Robinett, Virtual Environment Display System, *Proceedings of the 1986 ACM Workshop on Interactive Graphics*, October, 1986, 77-87.
33. S. S. Fisher, M. McGreevy, J. Humphries and W. Robinett, Virtual Interface Environment for Telepresence Applications, *Proceedings of the Human Factors Society 32nd Annual Meeting*, October, 1988.
34. F. P. B. Jr., Walkthrough - A Dynamic Graphics System for Simulating Virtual Buildings, *Proceedings of the 1986 ACM Workshop on Interfactive Graphics*, October, 1986, 9-21.
35. C. Brett, S. Pieper and D. Zeltzer, Putting It All Together: An Integrated Package for Viewing and Editing 3D Microworlds, *Proceedings of the 4th Usenix Computer Graphics Workshop*, October, 1987.
36. T. G. Zimmerman, J. Lanier, C. Blanchard, S. Bryson and Y. Harvill, A Hand Gesture Interface Device, *Graphics Interface '87*, May, 1987, 189-192.
37. J. D. Foley, Interfaces for Advanced Computing, *Scientific American*, October, 1987, 127-135.
38. *DataGlove Model 2 Users Manual*, VPL-Research, Inc., 1987.
39. I. E. Sutherland, Head-Mounted Three-Dimensional Display, *Proceedings of the Fall Join Computer Conference 33*(1968), 757-764.
40. J. Lanier, Personal Conversation , November, 1989.
41. D. E.Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley Publishing Company, 1989.