Giving CANDY to Children:
User-Tailored Gesture Input Driving an
Articulator-Based Speech Synthesizer

Randy Pausch and Ronald D. Williams

Computer Science Report No. TR-91-23
October 7, 1991

This work was supported in part by the National Science Foundation, the Science Applications International Corporation, the Virginia Engineering Foundation, the Virginia Center for Innovative Technology, and the United Cerebral Palsy Foundation.

Giving CANDY to Children:

User-Tailored Gesture Input Driving an Articulator-Based Speech Synthesizer

Randy Pausch & Ronald D. Williams
University of Virginia

Abstract

The CANDY Project (Communication Assistance to Negate Disabilities in Youth) seeks to provide a real-time speech synthesizer for disabled individuals, particularly non-vocal children with cerebral palsy. Existing speech synthesizers convert user input into discrete linguistic or phonetic symbols which are converted into sound. Complicated sentences must be created by concatenating lower level symbols, precluding real-time conversational speech. We have developed an articulator-based speech synthesizer which simulates the motion of the human tongue and produces the corresponding speech sounds in real time. The synthesizer is driven by two continuous input signals and non-disabled users can produce real-time speech with a joystick. Disabled users will drive the synthesizer via passive tracking of their body movements. Magnetic trackers attached to the user report their location and tailoring software allows each user to move the tracker in an optimal orientation and range. The user motion is then converted into the two continuous signals that drive the speech synthesizer. In this way, we hope to allow each child to compensate for their inoperative vocal tract by using their "best" set of muscles to operate a simulated vocal tract. The motion mapping software may also have future potential as a physical therapy aid.

to appear in: Communications of the ACM

Introduction

The CANDY project (Communication Assistance to Negate Disabilities in Youth) combines computer scientists, electrical engineers, speech pathologists, pediatricians, and occupational therapists. Our ten year long goal is to create a speech synthesizer for disabled individuals. We are currently in our third year of the project and have produced a set of interim results in both our speech synthesizer and gesture-based input. Our initial target population is children with cerebral palsy (CP), of whom there are approximately 400,000 in the United States alone [Connolly]. When adults are included in the count, the number of Americans affected by CP is approximately 700,000[UCP]. Persons with other disabilities, such as strokes and Alzheimer's disease, may also eventually benefit from our system.

CP can be broadly defined as brain damage that impairs motor control. Persons with CP are not necessarily mentally retarded, but they do exhibit wide variation in physical abilities. A significant portion of the CP 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," and this symbol is transmitted to a traditional text-to-speech synthesizer.

We have developed a speech synthesizer which is articulator-based. We simulate the position of physical articulators, such as the tongue, and synthesize the

sound that would be made by air passing through the vocal tract with real articulators in those positions. Surprisingly, understandable speech can be produced by moving the tongue and holding other articulators in fixed positions. Our current synthesizer models 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 representing tongue position, and we can produce limited speech in real-time using a joystick.

Our target population cannot use standard input devices to drive the synthesizer. Any approach using physical input devices would require the construction of individual hardware for each user, which is prohibitively expensive. Also, children with CP are often physically weak, making the control of any physical device tiring. As users fatigue, their efficiency with a particular device decreases and several different devices may be needed to accommodate various stages of fatigue. We avoid physical input devices by using magnetic trackers which report body motion. The only physical effort required of the user is movement of his or her body, and those motions are converted by software into continuous control signals for the speech synthesizer. This tailoring software allows us to create interfaces based on each user's individual abilities and will make it possible for those interfaces to adapt as the user fatigues.

Existing gesture recognition and speech synthesis systems are based on symbols. For example, several systems have attempted to synthesize speech using the deaf alphabet and/or a subset of American Sign Language as user input [Dramer, Loomis]. These systems attempt to understand or interpret gestures, and are commonly referred to as gesture recognition systems. Our approach is to map continuous data from one or more sensors to a set of continuous device control signals with no intermediate symbols. This should make it possible to produce fluid, real-time speech synthesis with smooth transitions from sound to sound.

An Articulator-Based Model of Speech

Existing augmentative communication devices convert symbols into synthetic speech through some form of text-to-speech synthesis. When the symbols represent very small linguistic units, such as sounds or single words, the user has greater conversational flexibility but must transfer many symbols across the human to machine interface. When the symbols represent larger linguistic units, the user enjoys significantly reduced demands from the interface at the cost of conversational flexibility. Both extremes and all intermediate compromises currently offered are frustratingly slow, with communication rates at least an order of magnitude slower than that of normal conversations.

The articulators in the human vocal tract exhibit many degrees of freedom, and the coordinated motion of these articulators produces fluid, conversational speech. Human speech is produced as the composition of continuous sounds. The character of these sounds at each instant in time is determined to a significant extent by the instantaneous configuration of the articulators in the vocal tract. The fact that the articulator motion is concerted is fortunate because it effectively reduces the number of parameters specifying the state of the vocal tract, providing hope that a control signal with severely limited degrees of freedom can be used to drive a continuous speech synthesizer.

Conversational speech synthesis is inhibited by the transfer of symbols across the user interface. While the mental production of speech may be principally a symbolic process, the generation of speech sounds in the vocal tract is physical and continuous. Of course, control of the human vocal tract involves complex coordination. Our hypothesis is that if speaking persons can control their complex vocal tract at conversational rates, then many non-speaking individuals should be able to control a simplified simulated vocal tract to synthesize speech at conversational rates. We expect that users may require a long time to control this unique speech prosthesis, as the learning process can best be compared to the steps required for vocal individuals initially learning to speak. One benefit of this approach is that children equipped with the speech prosthesis would be able to acquire speech using the normal developmental process. Existing speech synthesis methods require the children to first learn a symbolic language and then learn to drive the speech synthesizer with it.

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 [Coker, Haggard, Henke]. Our new limitation is that disabled individuals must be able to produce the articular control parameters in real-time.

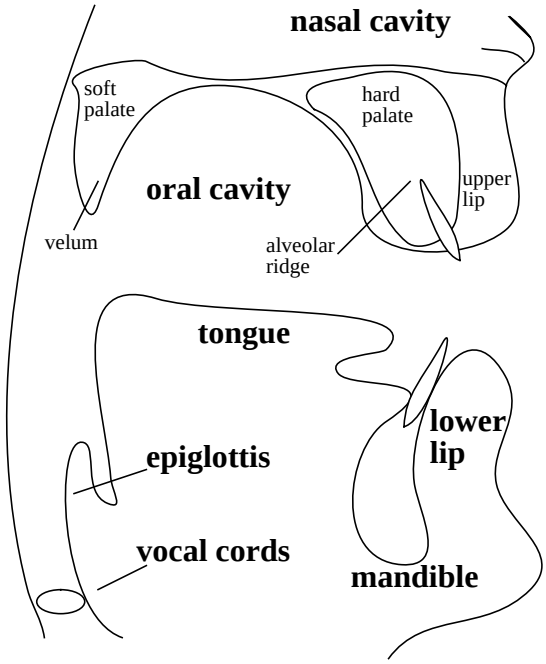


Figure 1: Human Vocal Tract

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. The articulators used to produce speech are shown in Figure 1. 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 [Zemlin].

Simplifying and Implementing the Model

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 capable of being 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. 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 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

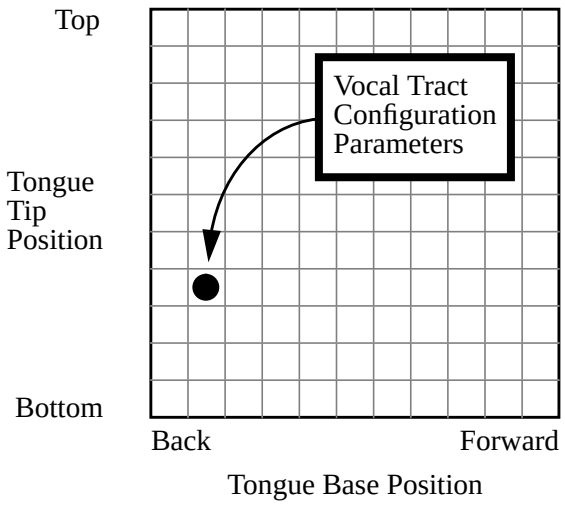


Figure 2: Two-D Tongue Position Grid

concatenating the /s/, /!EY!/, and /m/ units. Between these units, transients can occur that make the speech sound unnatural [Childers]. 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 [Girson]. 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.

Passive Tracking

When interfaces based on physical devices are problematic, an alternate approach is to passively track the user's body motions. Our general approach is to track user motions in three-dimensions and create custom projections to the two-dimensional tongue 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 CP 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 improvement and/or fatigue.

One alternative to tracking body motion is to track eye motion. 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.

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 recognition has been used for printed lettering, cursive handwriting, proofreader's symbols, and shorthand notation. In all cases, the approach is to convert the continuous motion of a stylus into a discrete token as input to a language-driven computation or process. 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 [Bolt, Buxton]. Systems have attempted to recognize static gestures for the deaf alphabet and motions for a subset of American Sign Language. All of these approaches are based on converting three-dimensional signals into a discrete stream of tokens.

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 [Furness]. More detailed tracking is performed in three dimensional drawing or sculpting applications [Schmandt], and virtual reality systems, where sensors attached to gloves [Foley] provide three-dimensional signals that are mapped into motions in synthetic worlds shown on traditional or head-mounted displays. These systems perform mappings from position and orientation information, but the mappings are significantly less complicated than those we propose.

The Experimental Setup

Our experimental setup is shown in Figure 3. One or two magnetic trackers are attached to the subject, at locations determined by a 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.

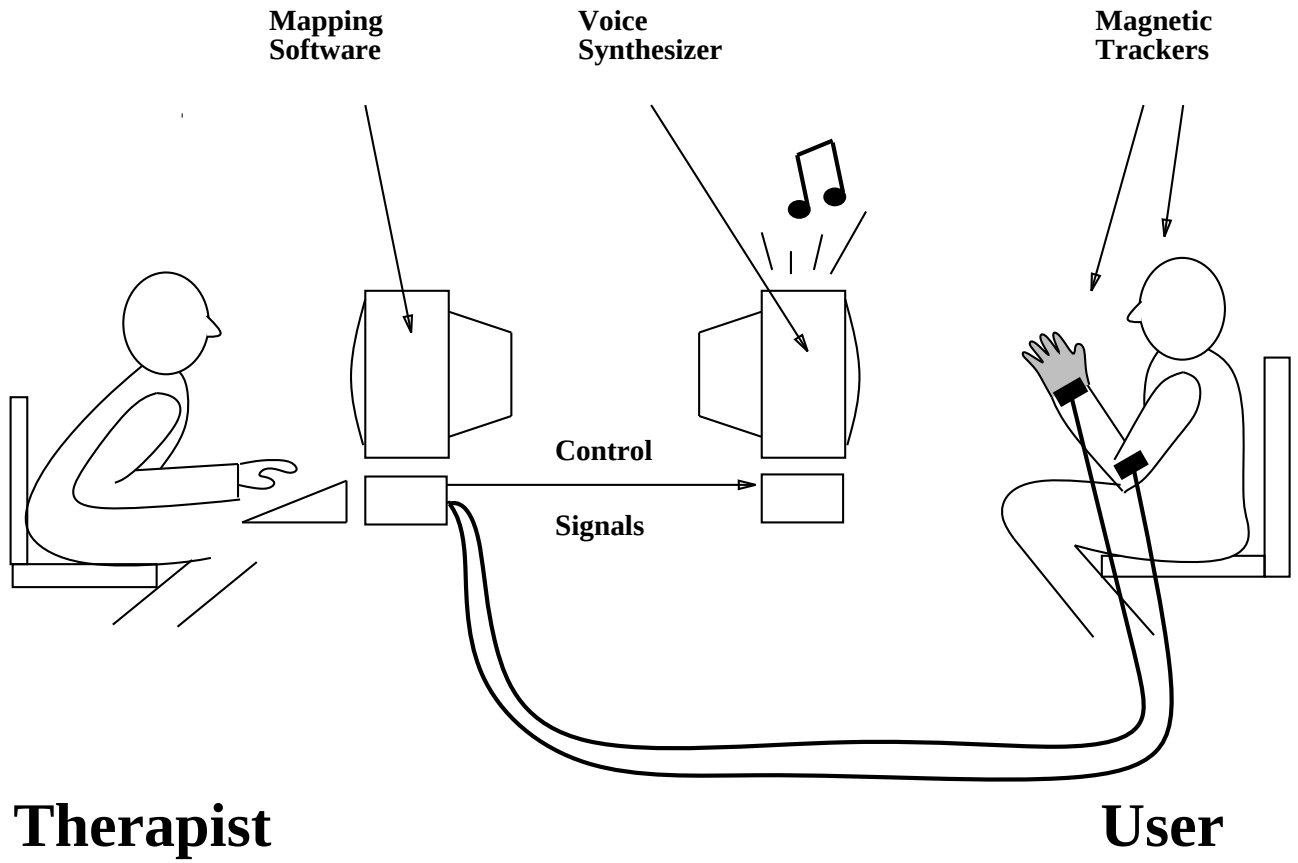


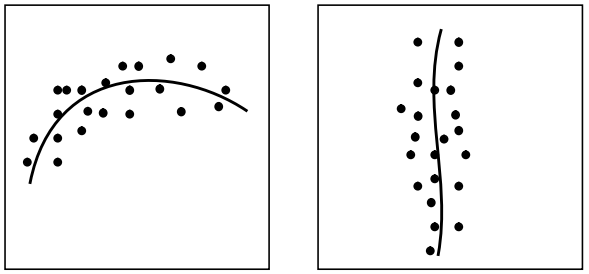
Figure 3: The Experimental Setup

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 using simpler one and two dimensional graphical applications with our disabled users.

Mapping Motion From 3D to 2D

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 in this paper, however these examples were chosen because their mappings are easily displayed geometrically.

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 vertical slider on a



User Moves Wrist By Pivoting Elbow

User Moves Wrist Forward and Backward

Figure 4: Target Curves

graphical display which can be moved up and down. 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 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.

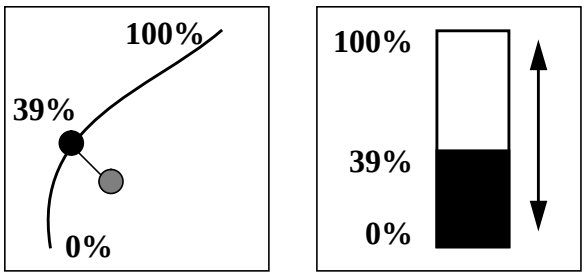
During the control phase the user moves the tracker along the target curve and we generate a linear control signal. One end of the curve indicates 0 percent of this signal and the other end indicates 100 percent. Intermediate positions along the target curve indicate intermediate signal values and the 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, the therapist can often 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.

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.

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 6.

The target surface is decomposed into a grid of planer sections that is then mapped into the grid for the two-



Tracker Space Video Feedback

Figure 5: Tracker Space to Device Space

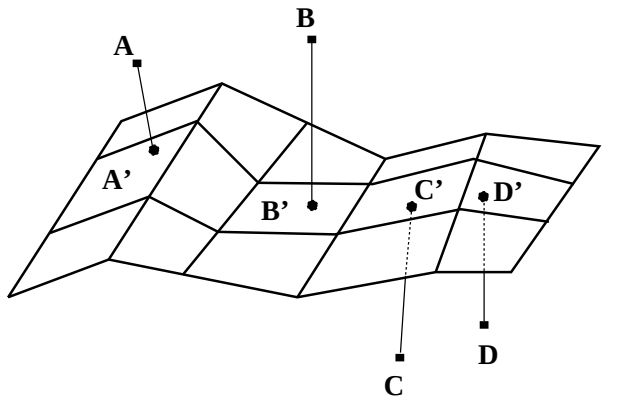


Figure 6: Mapping to a Target Surface

dimensional device signal, as shown in Figure 7. 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.

Creating the Target Curve

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 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 are initially concentrating on target curves and surfaces that can visualized by the therapist. We have created a low-cost "virtual reality" type head-mounted display using Private Eye displays[Becker] and a PowerGlove^{TM} which will eventually allow therapists to easily manipulate target surfaces [Pausch 91]. Later target curves and surfaces will be in spaces not easily visualized; in those cases, we will create the mapping entirely in software.

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

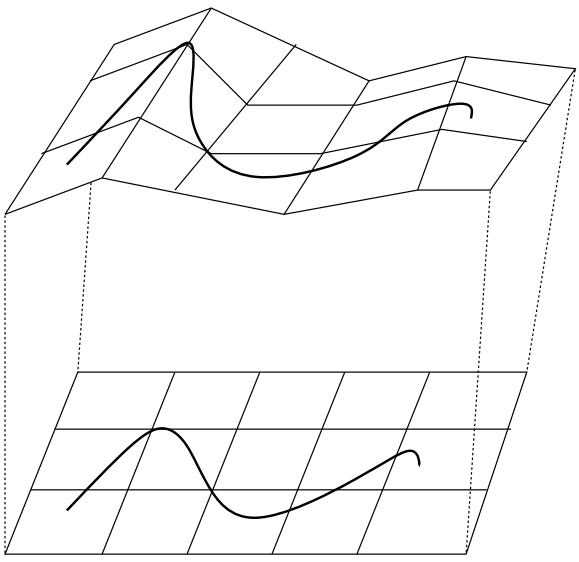


Figure 7: From Surface To Two Signals

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 [Pausch 90].

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.

Another potential use for the motion mapping software is as a physical therapy tool. When children use our system, they become tightly coupled with the action in our on-screen displays, similar to the behavior of video arcade game players. We hypothesize that if we were to stretch the target curve or surface during a session, a child would have to increase his range of motion in order to continue to perform well in the game. This places the child in a tightly coupled, self-motivating feedback loop, and may provide substantial advances over the current methods for physical therapy.

Conclusions

Able-bodied users can currently produce a variety of sentences in real-time driving the articulator-based speech synthesizer with a joystick. This requires a great deal of practice, but that is commensurate with our original goals. An active research area is finding grid layouts for speech sounds which are not direct mappings of tongue motion but provide a more usable grid. A major challenge is reducing the number of cases where one must travel through an undesirable sound in order to produce a sequence composed of sounds on opposite sides of that location. We are currently experimenting with both grid layout and "skipping" techniques, where rapid movement over a portion of the tongue grid allows one to avoid producing sound during the transition.

The user motion tracking software currently allows CP children to play a simplified pong video game where paddle motion is controlled by motion along a one-dimensional, user-specific target curve. We are currently performing clinical trials to measure the speed and accuracy of our CP children to evaluate their ability to eventually drive the synthesizer. In parallel, we are developing software to perform mappings using two dimensional target curves, and are exploring the possibility of using the mapping software as a tool for physical therapy.

References

[Becker] A. Becker, Design Case Study: Private Eye, Information Display, March, 1990.

[Bolt] R. Bolt, Put-That-There: Voice & Gesture at the Graphics Interface, Computer Graphics, 14,3 (1980), 262-270.

[Buxton] W. Buxton, E. Fiume, R. Hill, A. Lee and C. Woo, Continuous hand-gesture driven input, Proceedings of Graphics Interface '83, 191-195.

[Childers] 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.

[Coker] 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.

[Connolly] C. Connolly, Compensating for Cerebral Palsy: A Tailorable Mapping from Voluntary Movement to Synthetic Speech in Real Time, Master of Science Thesis, University of Virginia Department of Electrical Engineering, Charlottesville, VA, August, 1990.

[Dramer] J. Dramer, The Talking Glove in Action, Communications of the ACM 32,4 (April, 1989), 515.

[Foley] J. D. Foley, Interfaces for Advanced Computing, Scientific American, October, 1987, 127-135.

[Furness] T. A. Furness, Super Cockpit: Virtual Crew Systems, Armstrong Aerospace Medical Research Laboratory, 1988.

[Girson] A. Girson and R. Williams, Articulator-Based Synthesis For Conversational Speech, International Conference on Acoustics, Speech, and Signal Processing, April, 1990.

[Haggard] M. Haggard, Experience and Perspectives in Articulatory Synthesis, Frontiers of Speech Communication Research, London, UK, 1979, 259-274.

[Henke] 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.

[Loomis] J. Loomis, H. Poizner, U. Bellugi, A. Blakemore and J. Hollerbach, Computer Graphic Modeling of American Sign Language, Computer Graphics 17,3 (July 1983).

[Pausch 90] R. Pausch and R. D. Williams, Tailor: Creating Custom User Interfaces Based on Gesture, UIST ‘90: Proceedings of the Annual ACM SIGGRAPH Symposium on User Interface Software and Technology, October, 1990.

[Pausch 91] R. Pausch, Virtual Reality on Five Dollars a Day, Proceedings of the ACM SIGCHI Human Factors in Computer Systems Conference, April, 1991.

[Schmandt] C. Schmandt, Spatial Input/Display Correspondence in a Stereoscopic Computer Graphic Work Station, Computer Graphics 17,3 (July, 1983), 253-261.

[UCP] What Everyone Should Know About Cerebral Palsy, United Cerebral Palsy Association of Westchester County, Inc., 1985.

[Zemlin] W. R. Zemlin, Speech and Hearing Science, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1968.