GRASSP: Gesturally-Realized Audio, Speech and Song Performance

Bob Pritchard
School of Music
University of British Columbia
Vancouver, B.C. Canada
+1.604.822.3526
bob@interchange.ubc.ca

Sidney Fels
Dept. of Electrical Engineering
University of British Columbia
Vancouver, B.C. Canada
+1.604.822.5338
ssfels@ece.ubc.ca

ABSTRACT
We describe the implementation of an environment for Gesturally-Realized Audio, Speech and Song Performance (GRASSP), which includes a glove-based interface, a mapping/training interface, and a collection of Max/MSP/Jitter bpatchers that allow the user to improvise speech, song, sound synthesis, sound processing, sound localization, and video processing. The mapping/training interface provides a framework for performers to specify by example the mapping between gesture and sound or video controls. We demonstrate the effectiveness of the GRASSP environment for gestural control of musical expression by creating a gesture-to-voice system that is currently being used by performers.

Keywords
Speech synthesis, parallel formant speech synthesizer, gesture control, Max/MSP, Jitter, Cyberglove, Polhemus, sound diffusion, UBC Toolbox, Glove-Talk,

1. INTRODUCTION
The Gesturally-Realized Audio, Speech and Song Performance (GRASSP) environment is designed to synthesize speech and sound, and assist in real time processing of audio and video from real-time performer control through a mapping interface. For input, we created various input objects supporting a Cyberglove™, Polhemus Fastrak™ tracker, a custom-built left hand glove, and a footswitch, allowing the performer to accomplish all of this within a Max/MSP/Jitter environment.

The use of modified gloves, meta-gloves, and glove-like mechanisms for the control of synthesis and processing is not uncommon, as shown by such work as Waisvisz [1] and Sonami [2]. Indeed, gestural control is now common enough to warrant publications [3] dealing with the associated problems and solutions. Of particular interest in our project is the creation of a mapping and training component that quickly allows the user to provide examples for adapting the mapping as well as for learning the hand positions required for speech synthesis. This has implications for the expanded development and use of sophisticated controllers in the GRASSP environment.

2. CONTEXT
Speech synthesis is found across a wide spectrum of uses, ranging from children's games to sophisticated telecommunication applications. In general the implementations that are found in text-to-speech synthesis use either concatenative synthesis such as Festival[4], or else some form of text-to-speech converters such as DecTalk™ to drive the speech synthesizer. Text-to-speech is a powerful and successful method but it lacks the ability to improvise speech in real time, and it can be difficult to implement a variety of inflections for expressive purposes.

Glove-TalkII [5] is one example of a system that allows users to improvise speech and inflection, allowing for more natural sounding conversation with the user. Using a Cyberglove™, a Polhemus Fastrak™, a custom-made left hand glove, and a foot controller, the user is able to control the formants and noise components of speech through different hand and finger positions mapped to control parameters of a parallel formant speech synthesizer. Along similar lines, Cook [6] has created a vocal synthesizer that he controls with various bellows-based instruments including a concertina and an accordion.

One of the key features of Glove-TalkII is the ability of the user to provide examples to the system of hand gestures for different cardinal sounds. These examples were used to train the mapping between gesture and speech synthesis parameters using a neural network. We were motivated by this approach to create a similar, but expressively expanded environment for performers to control speech, sound and video parameters. In order to do this, we explicitly consider gestural control of sound and video parameters as a trainable mapping between inputs and outputs.

In Glove-TalkII, the training paradigm was done separately from the process of controlling speech. However, within Max/MSP, we are able to unify the objects that provide inputs and the objects that provide expressive output control through a generalized mapping interface that we call a *Trainer*. As Orio et al [7] point out, the mapping between inputs and outputs needs to be the focus of attention for understanding expressive control. Furthermore, Glove-TalkII illustrated that adaptive mappings can provide an effective way for a user to be able to train a mapping to suit their taste. Glove-TalkII started with an initial mapping that specified a reasonable starting point for producing speech from gesture based on a *hand-as-articulator* metaphor. The user's interpretation of that mapping determined the actual training data. In keeping with this approach, our training/mapping interface provides a mechanism to display the current mapping while also allowing the addition of new examples from the performer to adjust the mapping. Our current adaptation adjusts the centres of radial basis functions [8] using single examples. However, it is a simple matter to extend our

interface to include other machine learning techniques such as found in Glove-TalkII.

3. THE ENVIRONMENT

The GRASSP environment can be viewed as having three distinct yet interrelated components. These three parts and their roles are summarized as follows:

The Controllers: these consist of a Cyberglove™, a Polhemus Fastrak™, a custom left-hand glove, and a footswitch. This collection allows the user to generate control signals.

The Trainer: this consists of a Dictionary, a Recorder, Tracker, and a Mapper. These four Max/MSP/Jitter interfaces a) provide the user with models of hand positions associated with specific speech sounds, b) record the control data resulting from the user's attempts to imitate the position, c) show the user the approximate relation of the hand location to the target positions and d) provide the relation between the Controllers and the UBC Toolbox. The Mapper also contains the adaptation algorithm that uses the data from the Recorder.

The UBC Toolbox: this is a collection of Max/MSP/Jitter bpatchers, created by Hamel and Pritchard [9]. Included in the UBC Toolbox is the parallel speech synthesis engine of Glove-TalkII that has been recoded in Max/MSP and implemented as a bpatcher. We included this object so that performers can sing or speak using gestures.

Table 1: UBC Toolbox bpatcher examples

Category	Name/Description
Players	boxcar, filePlayer, fmPlayer, granulator, ksPlayer, pafPlayer, sampler, sfPlayer, vibePlayer, vSynthDB, vSynthDBFem,
Effects	chorus, combFilter, crossSynth, delay, multifilter, flange, fShifter, harmonizer, reverb, ringMod, vocoder
I/O	multipan4, multipan8, NAInterface, netSender, netReceiver, pan4, pitchTracker, recorder
Controls	cMatrix, control collections, dispatch, matrix, mixer, randGen
Jitter	chromaKey, crossfade, floatWindow, mathOps, messenger, motionDetector, videoGrabber, videoPlayer, writeToDisk

4. RELATED WORK

Fels and Hinton implemented the Glove-TalkII system based on the parallel-formant speech synthesizer of Rye and Holmes [10] which was developed at the Joint Speech Research Unit (JSRU) in the UK. Fels and Hinton created a gesture-controlled interface for their version of the JSRU synthesizer, and added an adaptive interface component to refine the user training and speech synthesis. The resulting system converts hand gestures to speech, using a gesture-to-formant model. The right hand location in 2D horizontal space controls the creation of vowel formants by mapping hand position to the amplitudes of the first two formants. Specific hand locations are identified as targets for specific cardinal vowels and their related formants. Normalized radial basis functions [8] provide a vowel landscape that produces cardinal vowel sounds as the hand approaches the target position and maintains it as the hand extends to extreme positions outside the main vowel space. The vertical position of the right hand controls pitch, and the right hand finger positions control the creation of all other speech sounds with the exception of eight stopped consonants. These eight consonants are triggered by contact switches on a left hand glove, and the overall amplitude of the system is controlled by a foot switch.

Hamel and Pritchard created the UBC Toolbox as a way of providing easily useable yet sophisticated modules. The collection contains over forty bpatchers, and all of the audio related ones use the same data and messaging protocol, and can be chained together or controlled via a bpatcher matrix.

5. PROJECT DESCRIPTION

5.1 Project Goals

The main goals of our project included the implementation of the JSRU synthesizer in the Max/MSP environment, the expansion of the speech synthesizer's capabilities to include the control of sound synthesis, sound processing, sound localization, and video processing, and the creation of a training system to ease the task of learning to synthesize speech. The Max/MSP/Jitter environment running on OS X was attractive for us since we have a great deal of experience with the environment, and almost all of the software development in the School of Music (synthesis methods, notation systems, teaching materials) is executed in OS X.

Secondary goals include using the gesture data to control kinetic sculptures and lighting systems, creating a direct correlation between gesture, sound, motion, and illumination. An additional goal is to develop a fully portable version of the environment by making the system self-contained.

5.2 Implementation

As our first demonstration of the utility of GRASSP, we have implemented a gesture-to-voice system based on Glove-TalkII. We use this example to illustrate the different components of GRASSP.

We divide the task of gesture-to-voice control into three areas: vowels, consonants, and stopped plosives. Vowels are considered to be those sounds made with the open mouth. Consonants are utterances created by constraining or restricting the air flow through the vocal passage, and stopped plosives are consonants in which the air flow is interrupted by the lips, tongue, teeth, palette, or some combination of those. In this example for Controllers we use data gloves, a tracking sensor, and a foot pedal for capturing gestures. The Trainer is set up to provide a mechanism for performers to see and add to the dictionary of gestures to vowel and consonant sounds. The Tracker allows performers to set the space of vowel control as well as to visually monitor the current location of their hand in that vowel space. The Trainer also provides the Mapper that generates the 11 control parameters of the JSRU speech synthesizer from the Controllers. Finally, we use a new version of the JSRU parallel formant synthesizer implemented in Max/MSP. This implementation forms a new member of our UBC Toolbox.

5.2.1 The Controllers

For the generation of control data we adopted the approach of Glove-TalkII, using a Cyberglove™, custom left-hand glove, a foot switch, and a Polhemus FastTrak™. It was necessary to write new Max/MSP objects to connect to the Cyberglove™ and FasTrak™, so glove_ini, glove, polhemus_ini, and polhemus objects were created. The _ini objects translate Cyberglove™ and Fastrak™ control messages and pass them on to the

standard Max/MSP **serial** objects to configure the respective hardware devices. The **serial** objects access USB (and RS232 ports) and pass the incoming data to the glove or polhemus object. The glove object supplies the 18 sensor readings required for synthesis, while the polhemus object supplies the X, Y, Z, Yaw, Pitch, and Roll coordinates of up to 4 trackers. We currently use only one tracker located on the back of the right hand wrist for our gesture-to-voice implementation.

Additional control data is generated by a foot switch and by a custom-made left-hand glove. The footswitch is a simple USB model that is intended to control the overall system volume as in Glove-TalkII. The left hand glove triggers are intended for the production of stopped plosives as in Glove-TalkII. It uses a qwerty keyboard chip for data control, and has nine touch points – two on each finger and one on the thumb. When the thumb contacts a touch point it triggers a bang message that can be used to produce one of eight stop consonant routines such a making a 'B' sound.

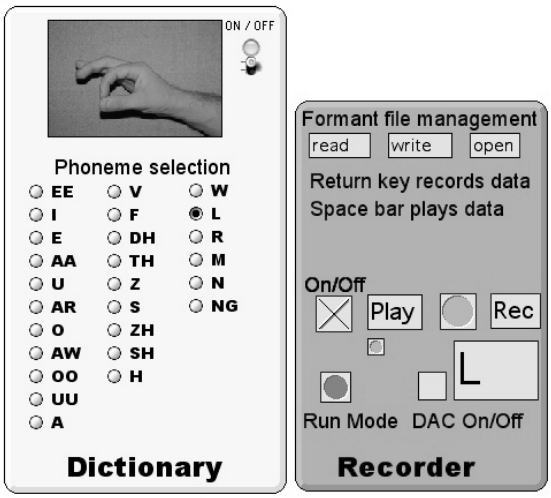

Figure 1: Dictionary and Recorder

5.2.2 The Trainer

The Trainer consists of a Dictionary, a Recorder, Tracker and an embedded Mapper. Figure 1 shows the Dictionary displaying the hand position for the phoneme 'L' for our implementation of a gesture-to-voice mapping. The Recorder is ready to accept training data being generated by the performer and write it into the data base for use by the Mapper. The Dictionary's illustrations are of a gloveless hand since this provides the trainee with greater detail of how to configure the hand. The Dictionary images provide a graphical view of the initial gesture vocabulary, and the images can be changed easily to accommodate different representations of the initial vocabulary. In our gesture-to-voice implementation, vowels do not require finger positions since they are determined by the horizontal location of the wrist, and the Tracker assists in creating the data necessary for their production.

The Tracker is shown in Figure 2. The Tracker displays a vowel quadrilateral based on the work of Petersen and Barney[11]. The small circle in the lower left of the quadrilateral is a cursor displaying the current position of the gloved right hand in relation to the Fastrak receiver. (The Tracker's coordinates are based on the user's frame of reference, with X increasing horizontally left-to-right, and Y increasing horizontally close-to-far.) The user is able to set the X, Y, and Z boundaries of the synthesis space by activating the appropriate button to record the location of the hand.

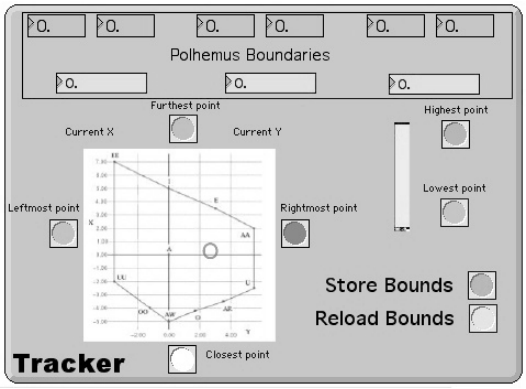

Figure 2: Tracker

Using the Trainer in our gesture-to-voice implementation, the performer sets the locations of eleven target vowels indicated in the quadrilateral. To do this, she selects a vowel from the left hand column in the Dictionary, moves her hand so that the cursor in the Tracker is over the chosen vowel, and enters that location using the Recorder. She also uses the Tracker to set her movement bounds of her vowel space. Likewise, the performer uses the Dictionary interface to store the 15 consonant gestures. The entire collection of data (consisting of the bounds of the space to be used in relation to the Fastrak receiver, the vowel locations, and the hand positions for each consonant) is known as an "accent". Each user develops and modifies his or her own accent, based on their personal preferences and their physical characteristics. Each vowel or consonant in an accent contains over 40 different floats and integers, as each contains all the sensor readings from the Cyberglove™ and FastTrak, eleven control values for the synthesis of that sound, and placeholders for future use.

The Mapper uses the accent to adjust the parameters of the relation between the Controllers and parameters of the output controls which in our case is the speech syntehsizer and other bpatchers in the UBC Toolbox. We currently adjust the centres of the normalized RBF functions using the single examples for each sound in the accent. However, different machine learning techniques may be used as well as a different training data collection scheme for this.

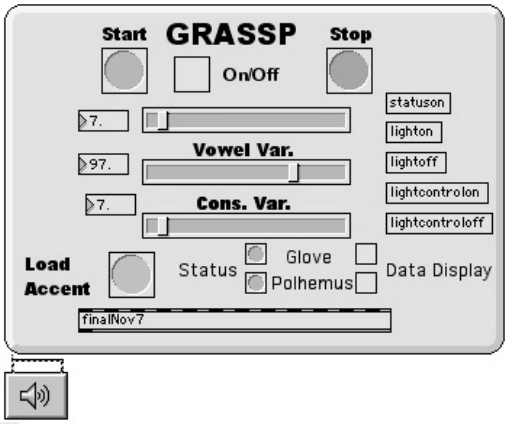

Figure 3: GRASSP bpatcher

5.2.3 The UBC Toolbox

As mentioned, the Toolbox currently contains over forty bpatchers. (See Table 1.) The bpatchers are capable of a variety of synthesis and processing duties in the audio and video domains, and are easily controlled by external inputs and/or and internal messaging, Our coding of the speech synthesis routines

in a bpatcher allows it to be integrated seamlessly into the processing and synthesis chain of the GRASSP environment. Figure 3 shows the GRASSP bpatcher. We focus on the implementation of the speech synthesizer here as it required significant effort and its implementation is not obvious.

5.3 The Speech Synthesizer

We implemented the JSRU parallel formant speech synthesizer [10] in Max/MSP as a bpatcher in the UBC Toolkit. The synthesizer accepts eleven control signals: FN, F1, F2, F3 -- the centre frequencies and amplitudes of vowel formants; AHF, A1, A2, A3 – the amplitudes of those frequencies; IV – the degree of voicing between voiced and unvoiced sounds; and F0 – the current fundamental frequency of the sound being synthesized. Ten of the control signals are low-pass filtered to eliminate artifacts caused by fast changes before being passed to the synthesizer. This is especially important since the control parameters are coming from the Mapper and may be quite discontinuous.

Following [10], inside the synthesizer there are six areas of control and synthesis: voiced excitation, unvoiced excitation, a mixer, gain units, formant resonators, and spectrum weighting filters. The relation of these areas is shown in Figure 4.

Voiced excitation is provided by digital representations of glottal waveforms, which are played using the cycle~ object. Unvoiced excitation uses the noise~ object.

The six channel mixer blends the voiced and unvoiced excitations according to the degree of voicing indicated by the IV control signal. The resulting signal is then sent to the six different formant resonators. At the same time, the centre frequency and bandwidth are calculated for each formant based on the input parameters and the data are passed to the formant resonators.

Each formant resonator contains a biquad~ object, and that object's output is phase matched before being passed on to the spectrum filter. All six channels are then combined and passed through the spectrum weighting filters which consist of a bank of five biquad~ objects. Additional heuristic adjustments to take into account spectral balancing, glottal energy equalization and special parameter couplings according to [10] are also implemented.

Each set of 11 input parameters are updated at 100Hz. The processing is handled effectively on a MacMini with spare capacity for managing the I/O. When voiced excitation is used, the system uses prerecorded glottal waveforms to drive the resonators. Collectively, the system synthesizes speech from the formant parameters supplied to the bpatcher.

5.4 Training of Gesture-to-Voice

Unlike the original Glove-TalkII, our gesture-to-voice implementation in GRASSP does not make use of neural networks to assist in teaching and refining the system. Instead, we use a simple method of setting vowel and consonant centres using a single gesture example for each sound. It is expected that the user will adapt to the interface: in much the same way that a performer must learn to adapt to a traditional instrument, GRASSP users must learn to adapt to this new instrument.

Training begins with the user creating a complete accent by recording the eleven target positions for vowels, the fifteen hand positions for consonants, and the spatial coordinates of the bounds.

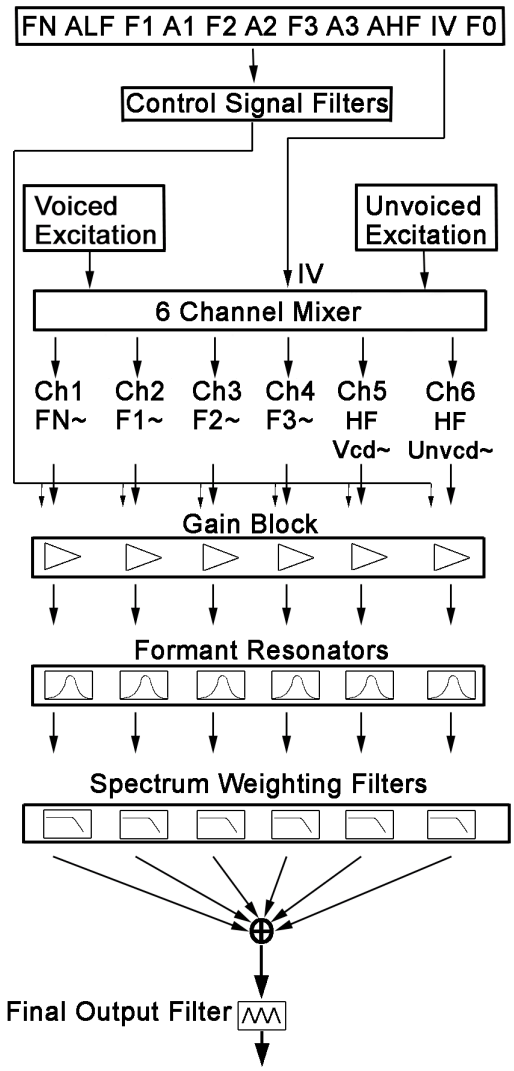

Figure 4: Block diagram of speech synthesizer

Once the accent has been established, training continues with the exploration of the vowel space, and then progresses to combining vowels with a variety of consonants. All users have expressed surprise at the discoveries they make about their own natural speech production, and to date it has not been an onerous task to spend time learning how to speak. Indeed, the enjoyment is shown as one test phrase currently used for comparing accents between users is "I need a beer!"

6. PERFORMANCE

In performance the audio output of the gesture-to-voice system in GRASSP can be fed directly to a sound system for diffusion as in a regular electroacoustic performance. However, we find it musically more interesting to add modules from the UBC Toolbox to the audio chain, thereby expanding the capabilities of the basic gesture-to-voice system.

For instance, using the horizontal coordinates of the different vowels to control one of the Toolbox's multipan4 or multiPan8 modules results in a one-to-one correlation between the vowel being synthesized and its localization within the concert hall. In effect, the audience is placed within the Petersen and Barney vowel quadrilateral. Or, from an articulatory phonetics point of view, it is as if the audience is inside the mouth at the quadrilateral approximates of the tongue position during vowel production. Because of the open architecture of the code and

the flexibility of Max/MSP, control signals can be pulled from any point in the synthesis chain and mapped onto other parameters. Thus, hand height can also control reverberation amount, different consonants can control different processes, and the left hand contact switches can be used to select different configurations via the matrix presets, or trigger samples or other processes.

Additionally, a very interesting performance configuration is possible by replacing the digitized glottal waveform with the sound of another live performer or with prerecorded samples. The result can be similar vocoder processing, with the strength of the effect dependent upon the richness of the substituted waveform.

7. FUTURE DIRECTIONS

Future work with GRASSP will include the recoding and refinement of the Max/MSP patches to overcome the limitations of sensitivity caused by the lack of precision in floats. Additionally, we may recode the entire synthesizer as a single Max/MSP object, making it easier for other users to include the object in their compositions. We will also be expanding the use of the interface via the UBC Toolbox to include graphic processing and the control of kinetic sculpture. The Essential Reality P5™ glove might also be of some interest once its yaw, pitch, and roll data is updated more frequently.

8. CONCLUSION

We have created an environment called GRASSP in Max/MSP that supports three main components for creating adaptive mappings between controllers and expressive sound and video. The components include objects for the Controllers, a Trainer and output control from the UBC Toolbox. To illustrate the effectiveness of GRASSP we have implemented a gesture-to-voice system based on Glove-TalkII since it requires all the components to be used. Our implementation required us to build controller objects for a Cyberglove™, a Polhemus Fastrak™, a custom left-handed glove and a foot pedal. We also ported the JSRU parallel formant speech synthesizer for our speech output. We created a unique training method to allow performers to adjust the mapping between gestures and voice. Using our gesture-to-voice system and the additional components in GRASSP, performers have begun to make new performance pieces. Thus, GRASSP has proven to be an effective environment for new forms of gestural control of expressive content.

9. ACKNOWLEDGMENTS

The GRASSP project is made possible by an Artist-Researcher Grant from the Social Sciences and Humanities Research Council of Canada, and is generously supported by the following University of British Columbia research units: Music, Sound, and Electroacoustic Technology (MuSET), the Media And Graphics Interdisciplinary Centre (MAGIC), and the Institute for Computing, Information, and Cognitive Science (ICICS).

10. REFERENCES

[1] Michel Waisvisz. The hands. In Proceedings International Computer Music Conference, pages 313-318, 1985.

[2] Sonami, L., http://www.sonami.net/lady_glove2.htm.

[3] Wanderly, Marcelo and Battier, Marc, eds. Trends in Gestural Control of Music. IRCAM – Centre Pompidou – 2000.

[4] http://festvox.org/festival/

[5] Fels, S. and Hinton, G., Glove-TalkII: A neural network interface which maps gestures to parallel formant speech synthesizer controls, IEEE Trans on Neural Networks, Vol 9, No. 1, pp. 205-212, 1998.

[6] P. Cook, "SPASM: a Real-Time Vocal Tract Physical Model Editor/Controller and Singer: the Companion Software Synthesis System," CMJ (Computer Music Journal), 17: 1, pp 30-44, 1992.

[7] Orio, N., Schnell, N., and Wanderley, M., Input Devices for Musical Expression: Borrowing Tools from HCI, New Interfaces for Musical Expression Workshop (NIME)'01 at ACM CHI 2001, Seattle, USA, April 2001.

[8] Fels, S., Using Radial Basis Functions to Map Hand Gestures to Speech, in Radial Basis Function Networks 2, New Advances in Design, ed. Howlett, R. J. and Jain, L. C., Physica-Verlag, pp. 59-101, 2001.

[9] http://www.opusonemusic.net/muset

[10] Rye, J.M. and Holmes, J.N. A Versatile Software Parallel-formant Speech Synthesizer. Joint Speech Research Unit Report No. 1016, 1982.

[11] Peterson, B. and Barney, H. "Control Methods Used In a Study of the Vowels," America, 24, 1952.