ARTIFICIAL SINGING WITH A WEBCAM MOUTH-CONTROLLER

Athanasios Hapipis and Eduardo Reck Miranda
Computer Music Research,
School of Computing, Communications and Electronicus
University of Plymouth
Plymouth PL4 8AA, United Kingdom
thanasis.hapipis@postgrad.plymouth.ac.uk, eduardo.miranda@plymouth.ac.uk

ABSTRACT

We present a device to control a singing synthesiser with mouth movements captured with a mini web camera. The device comprises a mini web camera connected to a computer, which extracts a number of mouth movement parameters. These parameters are sent out as MIDI messages to another computer running a formant synthesiser, which produces singing according to the movements of the mouth of the subject in real-time. The paper gives a technical explanation of the vision system, the synthesiser and the control parameters. Our main motivation with this research is to enable people with speech or voice disorders to engage in singing activity.

1. INTRODUCTION

New technological advancements result in creating new levels of interaction between a human and a computer (machine), therefore reinforcing their relationship. The levels of this interaction are supported by the use of appropriate hardware and/or software interfaces for controlling the interaction model. These are concepts and terms brought forward by the Human-Computer Interaction (HCI) field of Computer Science and are indisputably applicable for human-computer interaction in the Arts and indeed Music. However, when expressiveness and creativity come into question, the interfaces developed according to strictly standard HCI principles can be, to some extent, restricted.

The exploration of creativity in human-computer (and human-machine in general) interaction is closely coupled with the evolution of music: a musical instrument can be described as an interactive interface or as a data controller [1]. Music interfaces naturally tend to explore the factors of expressiveness and creativity of the particular music interaction model they control.

This paper describes the development of a music controller that employs computer vision to extract mouth shape parameters in order to control a formant synthesiser.

2. BACKGROUND

2.1. Real time performance controllers

A number of researchers have underpinned the difficulties of developing real-time performance controllers for synthesized singing voice. Most of these difficulties are related to the vast number of parameters to be controlled and functional limitations brought forward by the technology used [2].

Real-time controllers such as Perry Cook's *SqueezeVox*, which is a modified accordion, and *VOMID*, which is a customized keyboard synthesiser, are very interesting "instrument-inspired" (in the sense that it uses the control surface of well-known standard musical instruments) interfaces used for expressive singing synthesis [2]. The choice of these musical instruments as controllers is well justified by the fact that they embody control features that are suitable for singing synthesis, particularly *SqueezeVox*; e.g., pitch, breathing and articulation.

Because our main motivation is to enable people with speech or voice disorders to be able to engage in singing activity, we are interested in building upon these developments by adding a control surface that is more identical to our natural voice "controller": i.e., the mouth.

2.2. Acoustic mouth-controllers

Since voice is naturally controlled by the mouth and affected by other internal organs such as the vocal cords, the oesophagus, the tongue, and so on, expressive controllers were developed that made use of the oral cavity to control singing synthesis and sound effects. These types of controllers are known as acoustic mouth controllers. *TalkBox* and *Tongue 'n' Groove* [11] are excellent examples of such controllers. These controllers maximize expressiveness due to the fact that they resemble the interaction model of the mouth. On the other hand, maximising expressiveness reduced the controller's usability. *TalkBox* requires the performer to hold a small speaker inside the mouth and *Tongue 'n' Groove* uses an ultrasound device held just below the jaw monitoring tongue movement inside the oral cavity.

2.3. Vision-based controllers

Vision-based controllers make use of computer vision techniques to distinguish colours, shapes and motion. Natural singing voice is relatively more closely coupled with the movements of the mouth than with the other organs responsible for producing voice. The development of a vision-based controller that uses mouth shape parameters to produce singing voice is therefore appropriate as a good balance between expressiveness and usability.

The *Mouthesizer* is an excellent example of a vision-based controller [7]. It uses a mini head-worn camera, which interprets mouth shapes and produces control messages for a MIDI device generating sounds or sound effects. The *Mouthesizer* has been used for three musical applications: guitar effects, keyboard and sequenced loops – but not for singing synthesis.

Such vision-based controllers seem to maximize the expressiveness of interactive performances more than the acoustic mouth-controllers described earlier. The development of vision-based controller, however, falls into many difficulties regarding the choice of the technologies to be used in order to result in efficient, successful and robust data control without ignoring the expressiveness factor of such controllers [8, 10]. Appropriate vision processing algorithms should be chosen including appropriate choice of methodology for mouth tracking and parameter extraction together with suitable parameter conversion and representation to control the software singing synthesiser. All these issues have been carefully thought and addressed during the early development stages of our device and are briefly discussed in the following sections of this paper.

3. THE SYNTHESISER

The singing synthesiser is a source-filter formant synthesiser (Figure 1). A source-filter synthesiser is based on the insight that the production of vocal sounds can be simulated by treating it as the generation of some type of raw source sound which subsequently passes through a filter arrangement [9]. In humans, the raw sound source would correspond to the outcome from the vibrations created by the vocal folds and the filter arrangement to the vocal tract [5].

The implementation of the filter arrangement is based upon measurements of the human vocal tract. In general, the vocal tract is considered as a tube (with a side-branch for simulating the nose) sub-divided into a number of cross-sections whose individual resonance is simulated by a filter. The outcome $\Phi(f)$ of the source-filter synthesiser can be characterised in the frequency domain as follows:

$$\Phi(f) = S(f)\,\Delta(f)$$

where S(f) is a source signal and $\Delta(f)$ is a linear transfer function defined by the filter arrangement.

Given an input signal $x(n\vartheta)$, such as a pulse train, and a constant ϑ equal to the inverse of the sampling rate, the filter arrangement is composed by a combination of digital resonators of the following form:

$$\delta(n\vartheta) = Ax(n\vartheta) + B\delta(n\vartheta - \vartheta) + C\delta(n\vartheta - 2\vartheta)$$

where $\delta(n\vartheta - \vartheta)$ and $\delta(n\vartheta - 2\vartheta)$ are the previous two samples of the output $\delta(n\vartheta)$. The values of A, B and C are specified according to the resonance centre frequency F_c and bandwidth W values of the desired formant, as follows:

$$C = -e^{(-2\pi W\vartheta)}$$

$$B = 2e^{(-\pi W\vartheta)}\cos(2\pi F_C\vartheta)$$

$$A = 1 - B - C$$

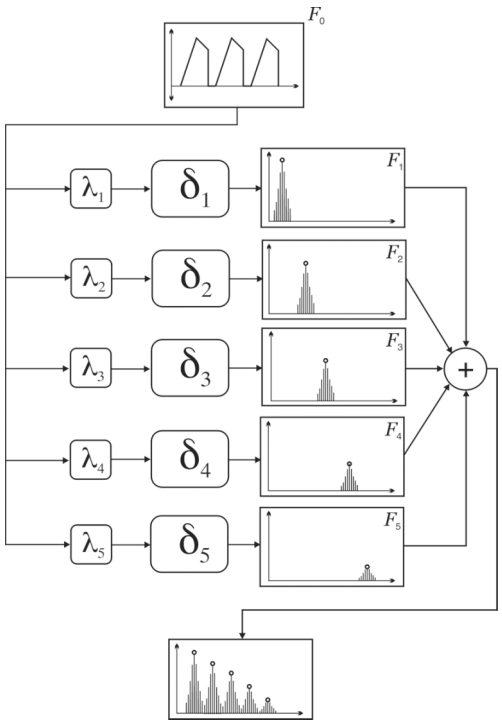

Figure 1. The singing synthesiser's architecture.

The synthesiser uses five of such digital resonators in parallel, each of which produces one formant; there is no provision for nasals in this model. The singing sound therefore results from the addition of the signals produced by each resonator δ_n scaled by an attenuation coefficient γ_n that defines the power of each formant in the overall spectrum:

$$\Phi(t) = \delta_1(t)\gamma_1 + \delta_2(t)\gamma_2 + \ldots + \delta_6(t)\gamma_6.$$

The synthesiser features an articulatory model whereby five resonance formant frequencies (F_1, …, F_5) are represented in terms of three parameters:

a) Position: position of the tongue in the front-back dimension (zero = maximum front and 1 = maximum back)

b) Height: height of the highest point of the tongue. This controls the "openness" of the sound (zero = maximum openness and 1 = maximum closeness)

c) Rounding: rounding of the lips (zero = lips are fully spread and 1 = lips are very rounded)

The equations for F_1 and F_2 of the articulatory model are provided in the Appendix 1. The synthesiser also features a vibrato mechanism, which is implemented by modulating the pitch frequency (F_0) with a non-linear Low-Frequency Oscillator (LFO). The control parameters of the synthesiser therefore are: duration (of the synthesised sound), loudness, vibrato, pitch, position, height and rounding.

4. THE VISION SYSTEM

The vision system consists of a mini web camera that reads and analyses mouth movement using appropriate computer vision algorithms. The various extracted parameters are converted into MIDI messages.

4.1. Design Issues

Designing a vision tracking device or software component requires the establishment of some ground design principles. Deciding what face features to track and how to achieve it is a taunting task that depends heavily on testing and refinement. Some studies in the past have identified methodologies and techniques that can be used for accurate automatic lip tracking in the context and scope of their research [3, 6, 7, 8].

A number of researchers proposed lip reading by tracking facial features such as the skin colour, the eyes, lip corners and the nostrils [3, 10]. These techniques for lip reading are appropriate and successful in trying to locate the mouth area but are also computational expensive because of the tracking of many different features simultaneously. The *Mouthesizer* [7, 8] took these issues into account and restricted the tracking task to the mouth area only. This approach to lip tracking, which is the one we adopted in our device, is independent from the performer's gesture and head position, adding more flexibility to the interaction model.

4.2. Mouth Movement Tracking

Mouth movement tracking by means of computer vision can be achieved by various different techniques that take into account lips and mouth characteristics such as the colour of the lips, the lips contour and movement [3, 10]. However, these characteristics are unique for each performer and are unavoidably influenced by lighting conditions affecting the overall performance of the controller.

We adopted a different approach for mouth tracking by measuring the shape of the mouth and extracting parameters according to the intensity of the colour of the pixels inside the visible mouth area. Therefore there is no need for a face tracking system to analyse facial characteristics in order to compute the position of the mouth area. With our approach we do not need to consider the different mouth and lip characteristics of different performers, which adds more flexibility to the device. Once the mouth area is found, then further vision analysis and motion analysis is applied to extract the control parameters (Figure 2).

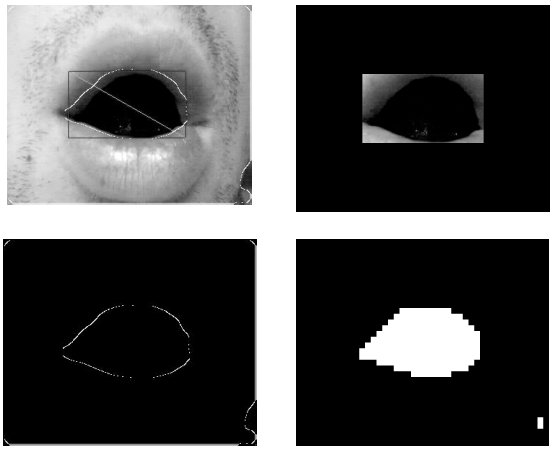

Figure 2. Tracking the mouth area.

4.3. Extracted Parameters

During the tracking process several parameters about the shape and motion of the mouth can be extracted from the visual input signal.

Currently, the system extracts the following parameters (Figure 3):

- The width of the mouth opening
- The height of the mouth opening
- The opening perimeter
- The diameter of the mouth opening
- The degree of motion
- The rounding factor

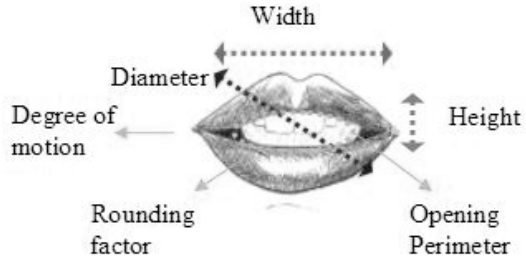

Figure 3. Extracted parameters

The values of these parameters are filtered for consistency and accuracy and then converted onto suitable MIDI messages to control the singing synthesiser.

We are aware of the advantages of working with a large number of parameters together with appropriate parameter mappings for creative and expressive performance behaviours [4]. Nevertheless, we decided to limit ourselves to six parameters; these proved to be enough for our purposes.

5. MAPPING

There are two categories of parameters and variables that control the synthesiser. The fist category corresponds to low-level parameters of the synthesiser architecture: duration, loudness, vibrato and pitch. The second category corresponds to the parameters of the articulator: position, height and rounding.

The mapping between the vision system and the synthesizer is a crucial issue for the design of musical controllers [4]. One-to-one parameter mappings allow users to better understand the interaction model because there is a strong relationship between cause and effect. On the other hand, complex parameter mappings (i.e., one-to-many or many-to-one) result in more complex and richer interaction models, but the relationship between cause and effect is not so clear in this case.

We tried many combinations including one-to-one, multiple-to-one and one-to-multiple associations; for example, an interesting behaviour was obtained by mapping the diameter of the mouth onto the whole set of articulatory parameters simultaneously. The mapping that seemed to work more intuitively is the one-to-one mapping shown in Table 1. This is the default mapping, but we implemented a facility that allows the use to change the mapping on the fly, if necessary.

Vision System	Synthesiser
Width (mouth opening)	Position
Height (mouth opening)	Height
Perimeter	Pitch
Diameter	Loudness
Motion	Vibrato
Rounding	Rounding

Table 1. The default parameter mapping.

As for the duration, the sound starts when the mouth opens and stops when the mouth closes.

6. IMPLEMENTATION TOOLS

The singing synthesiser was implemented in Csound and Max/MSP [12, 13] and runs on a Macintosh computer. The core of the Csound orchestra code is provided in Appendix 2. The vision system was implemented in EyesWeb [14]. EyesWeb's computer vision libraries are based on the Intel's Open Computer Vision (OpenCV) libraries, which are C/C++ algorithms and functions, freely available to researchers.

7. CONCLUSION

The development of an interactive music controller brings up many different issues that need to be taken into account, including the technology employed for interfacing and the interaction model between the performer and the system [1, 4]. Attention was paid to previous work done in the area of real time interactive controllers for voice synthesis and in the area of computer vision processing with valuable lessons learned during the research stage.

The developed mouth-controller attempts to maximise expressiveness and creativity and enables the performer not only to operate the voice synthesiser with efficient control but can also enhance the performance levels by having flexible parameter mappings that can be modified on the fly. Users and performers can experiment with different control mappings that can lead to interesting performance behaviours.

As stated earlier, the main motivation of this research is to enable people with speech or voice disorders to engage in singing activity. We are currently implementing a more robust device to make it available to professionals working with assistive technology for disability. This new device will replace the formant synthesiser by a physical model in order to render the output more realistic. Furthermore, we are considering replacing the MIDI communication protocol by OSC [15], as OSC provides a larger bandwidth than MIDI.

8. REFERENCES

[1] Bongers, B. "Physical Interaction in the Electronic Arts: Interaction Theory and Interfacing Techniques for Real-Time Performance", *Trends in Gestural Control of Music*, IRCAM, France, 2000.

[2] Cook, P. "Real-Time Performance Controllers for Synthesized Singing", *Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-05),* Vancouver, Canada, 2005.

[3] Goecke, R., Millar, B., Zelinsky, A. and Ribes, R. "Automatic Extraction of Lip Feature Points", *Proceedings of the Australian Conference on Robotics and Automation ACRA2000*, 2000.

[4] Hunt, A., Wanderley, M. and Paradiso, M. "The importance of parameter mapping in electronic instrument design", *Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-02),* Dublin, Ireland, 2005.

[5] Klatt, D. H., "Software for a cascade/parallel formant synthesiser", *Journal of the Acoustical Society of America*, Vol. 67, No. 3, pp. 971-995, 1980.

[6] Lyons, M. and Tetsutani N. "Facing the Music: A Facial Action Controlled Musical Interface", *Proceedings of the Conference on Human Factors in Computing Systems ACM CHI 2001,* Seattle, Washington, 2001.

[7] Lyons, M., de Silva, G. and Smyth, T. "A Novel Face-tracking Mouth Controller and its Application to Interacting with Bioacoustic Models", *Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-02)*, Hamamatsu, Japan, 2002.

[8] Lyons, M., Haehnel M., Tetsutani N. "Designing, Playing and Performing with a vision-based Mouth Interface", *Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-03),* Montreal, Canada, 2003.

[9] Miranda, E. R., *Computer Sound Synthesis for the Electronic Musician*. Oxford (UK): Focal Press, 1998.

[10] Tian, Y., Kanade, T. and Cohn, J. "Robust Lip Tracking by Combining Shape, Color and Motion", *Proceedings of the 4th Asian Conference on Computer Vision (ACCV'00)*, 2000.

[11] Vogt, F., McGaig, G., Ali M. and Fels S. "Tongue 'n' Groove", *Proceedings of the 2002 Conference on New Interfaces for Musical Expression (NIME-02),* Dublin, Ireland, 2005.

[12] http://www.csounds.com/matt/csound~/ (Last visited 09 Jun 2005).

[13] http://www.cycling74.com/index.html (Last visited 09 June 2005).

[14] http://www.eyesweb.org (Last visited 09 June 2005).

[15] http://www.opensoundcontrol.org/what_is_osc/ (Last visited 09 June 2005).

9. APPENDIX 1: THE ARTICULATORY MODEL

The equation for representing the first two formant frequencies of a sound produced by our synthesiser in function of position (p), height (h) and rounding (r) are given as follows:

$$F_1 = ((-392 + 392r)h^2 + (596 - 668r)h + (-146 + 166r))p^2 + ((348 - 348r)h^2 + (-494 + 606r)h + (141 - 175r))p + ((340 - 72r)h^2 + (-796 + 108r)h + (708 - 38r))$$

$$F_2 = ((-1200 + 1208r)h^2 + (1320 - 1328r)h + (118 - 158r))p^2 + ((1864 - 1488r)h^2 + (-2644 + 1510r)h + (-561 + 221r))p + ((-670 + 490r)h^2 + (1355 - 697r)h + (1517 - 117r))$$

10. APPENDIX 2: CSOUND ORCHESTRA FILE

```
<CsoundSynthesiser>
<CsInstruments>
sr=44100
kr=2205
ksmps=20
nchnls=2
;-------------
; Score variables
; p3        duration
;-------------
instr 1
;-------------
kamp=ampdb(90)
kfund     invalue "fund_freq"
kfc1      invalue "form_freq_1"
kbw1      invalue "form_band_1"
kfc2      invalue "form_freq_2"
kbw2      invalue "form_band_2"
kat2      invalue "att_freq_2"
kfc3      invalue "form_freq_3"
kbw3      invalue "form_band_3"
kat3      invalue "att_freq_3"
kgain     invalue "gain"
kvbrate   invalue "vibrato_rate"
;-------------
; Vibrato-jitter unit
;-------------
krnd1     randi .02, .05
krnd2     randi .02, .111
krnd3     randi .02, 1.219
kjit=(krnd1+krnd2+krnd3)*kfund
kvib      oscil kfund*.26, kvbrate, 1
kf0=kfund+kvib+kjit
;
ktime     times
outvalue  "perform_time", ktime
```

```
;-------------
; Sinusoidal voicing source
;-------------
knh=int(11025/kfund)
apulse buzz kamp, kf0, knh, 1
alpf1 reson apulse, 0, (kf0*2)*1.414, 1
alpf2 reson alpf1, 0, (kf0*4)*1.414, 1
asinu balance alpf2, apulse
;-------------
; Parallel filters
;-------------
af1 reson asinu, kfc1, kbw1, 1
af2 reson asinu, kfc2, kbw2, 1
af3 reson asinu, kfc3, kbw3, 1
atot balance (af1+(af2*(kat2*.01))+(af3*(kat3*.01))), asinu
;-------------
; Envelope
;-------------
kenv        linseg 0, p3*.15, 1, p3*.25, 1, p3*.4, 0
aout=atot*kenv*kgain
outs        aout, aout
endin
</CsInstruments>
```