INTERSPEECH 2016
September 8–12, 2016, San Francisco, USA

SERAPHIM
A Wavetable Synthesis System with 3D Lip Animation for Real-time Speech and Singing Applications on Mobile Platforms

Paul Yaozhu Chan[1], *Minghui Dong*[1],
Grace Xue Hui Ho[2], *Haizhou Li*[1]

[1]Human Language Technology Department, Institute for Infocomm Research, A*Star, Singapore
{ychan, mhdong, hli}@i2r.a-star.edu.sg
[2]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
gho004@e.ntu.edu.sg

Abstract

Singing synthesis is a rising musical art form gaining popularity amongst composers and end-listeners alike. To date, this art form is largely confined to offline boundaries of the music studio, whereas a large part music is about live performances. This calls for a real-time synthesis system readily deployable for onstage applications.

SERAPHIM is a wavetable synthesis system that is lightweight and deployable on mobile platforms. Apart from conventional offline studio applications, SERAPHIM also supports real-time synthesis applications, enabling live control inputs for on-stage performances. It also provides for easy lip animation control. SERAPHIM will be made available as a toolbox on Unity 3D for easy adoption into game development across multiple platforms. A readily compiled version will also be deployed as a VST studio plugin, directly addressing end users. It currently supports Japanese (singing only) and Mandarin (speech and singing) languages. This paper describes our work on SERAPHIM and discusses its capabilities and applications.

Index Terms: singing synthesis; speech singing; real-time; live; mobile platform; talking head; multimodal synthesis; lip synchronization

1. Introduction

Since Kratzenstein's first recorded attempt at mechanical voice reproduction in 1779 and Stewart's formant synthesizer in 1922, the reproduction of the human voice continues to carry significant research interests[1, 2]. Singing synthesis, in particular, continues to fascinate, entertain, and challenge researchers, composers and end-listeners alike[3].

Table 1 lists some of the prominent works in the area loosely categorized into four methods. Formant synthesis models the human voice with an oscillator and series of resonant filters [4, 5, 7]. Physical-modeling synthesis is based on a physical model of the human vocal tract[8]. Concatenative, or unit selection synthesis selects appropriate phone units from a database and concatenates them[12]. Parametric systems build a database of time-models of singing features and invoke them during synthesis[16, 14].

Of the four methods, formant synthesis gives the musician best control over the sound. Concatenative synthesis remains the best quality and least computationally intensive. A lot of research is focused around parametric synthesis, and, with the recent advancements in LSTMs, it is anticipated to soon overtake concatenative synthesis in terms of quality. High quality physical models, however, still remain elusive due to the complexity of building and ambiguity of fitting them.

Wavetable synthesis had been widely used in music ever since technology deemed it practical to store wave data[17]. It is the basis of compound synthesis, vector synthesis, and linear arithmetic synthesis[18]. When applied to singing synthesis, it may be viewed as a special case of concatenative systems where minimalistic coverage is made possible with exhaustive offline processing of the synthesis database. This makes wavetable synthesis especially suitable for real-time[1] applications on mobile processors.

Existing singing synthesis systems target mostly offline applications with only a small handful[7, 11] catering to real-time applications. Music, being a performing art form, places significant demand for real-time systems. Table 2 lists 2 of the most notable real-time singing synthesizers today together with SERAPHIM. As shown, in the table, SERAPHIM is the only system that uses dynamic gestures and is capable of synthesizing both vowels and consonants in real-time to the sub-frame level.

Table 2: Real-time Singing Synthesis Systems

	VOCALOID Keyboard[10]	**Cantor Digitalis**[4]	**SERAPHIM *Live!***
Developers	Yamaha, Universitat Pomeau Fabra	LIMSI	Institute for Infocomm Research
Interface	Static Buttons	Static Gestures	Dynamic Gestures
Real-time Capability	Syllable level	Sub-Frame level: Vowels only	Sub-Frame level: Vowels and Consonants

SERAPHIM is an extremely lightweight wavetable synthesis system that hogs a very small memory footprint and is designed for real-time implementations on mobile platforms. When compared with existing systems, our results show that

[1]Real-time sometimes refers to the systems that synthesize voices upon complete receipt of an input sequence. Throughout this paper, those will be disambiguated as runtime systems. Real-time systems, here, refer to systems that synthesize voices that reflect the state of the current input at every instant. Hence, for example, Stephen Hawking's text-to-speech system is not regarded as real-time in this context.

http://dx.doi.org/10.21437/Interspeech.2016-484

Table 1: Existing Systems. This table lists some prominent complete systems published. (The year column states the earliest available year of release or publication, not the most recent publication.)

Year	Author	Affiliation	Work	Languages	Method
1977	Larsson	KTH	MUSSE[4]	Various	Formant Synthesis
1983	X Rodet, Y Potard, J-B Barriere	MIT	CHANT[5, 6]	English	Formant Synthesis
2004	C. d'Alessandro, L. Feugere, Et al.	LIMSI, Virsyn	Cantor[7]	French, English, German	Formant Synthesis
1991	P R Cook	Stanford University	SPASM[8]	English, Latin, Greek	Physical Modeling
1996	M Macon	Georgia Tech, Texas Instruments	Lyricos[9]	English	Concatenative Synthesis
2004	H Kenmochi, X Serra, J Bonada	Universitat Pompeu Fabra, Yamaha	Vocaloid[10, 11, 12]	Various	Concatenative Synthesis
2008	Ameya / Ayame	-	UTAU[13]	Various	Concatenative Synthesis
2010	K Oura, Et al.	Nagoya Institute of Technology	SinSy[14, 15]	Japanese	Parametric Synthesis
2013	J-Y Cheng, Y-C Huang, C-H Wu	NCKU	- [16]	Mandarin	Parametric Synthesis

SERAPHIM performs 26.04% better than Cantor Digitalis and 18.54% better than Vocaloid in subjective listening tests.

This section provided an introduction to the background and motivation to SERAPHIM. The next section will cover methodology and Section 3 will follow by implementation. Finally, 4 presents our experiment and results, and Section 5 closes with the conclusion.

2. Methodology

2.1. Model Database

2.1.1. Syllable Structure and Phonetic Background

In Mandarin, syllables are made up of an initial consonant phoneme and a final that is compounded of several phones which might include a consonant coda. There are up to about 22 initials and 35 finals, depending on how they are being classified. SERAPHIM classifies them into 4 groups according to their initials for syllable modeling, as follows:

- Nulls - this is a special group of syllables without an initial. They begin with the first phoneme of the final which is a vowel and may be a suprasegmental.
- Voiced Consonants - this consists of all syllables whose initials are voiced.
- Unvoiced Consonants - this loosely groups all syllables whose initials are unvoiced or semi-voiced except those in the following group.
- Sibilances - this loosely groups all syllables whose initials are long, completely unvoiced sibilances and fricatives.

Similarly, in Japanese, syllables are also made up of an initial consonant phoneme and a final, except in Japanese, the final is almost always a single vowel. Suprasegmentals (e.g., 'Y' and 'W') are regarded as initials rather than compounding them as part of the final. SERAPHIM, hence, classifies syllables similarly for Japanese, but the Null group is not applicable.

Since the system is designed with real-time implementation in mind, as much of the computations are performed offline as is feasible. Two model databases are used in SERAPHIM and these have to be built offline. These are the syllable model database and the phone model database.

2.1.2. Phone Model Database

Phone models describe phone spectra at runtime, and since SERAPHIM is a light-weight time-domain system, the phone model is essentially a wavetable sequence of the phone.

Unlike conventional concatenative systems, where diphones are the primary unit for concatenation, in SERAPHIM, phones and biphones are used. Each syllable of the Voiced Consonants group has its initial consonant modeled by a phone model. Each syllable of the Unvoiced Consonants group has its initial consonant modeled together with its first vowel by a biphone model. Currently, each syllable of the Sibilances group has its initial consonant modeled together with its first vowel by a biphone model as well, similar to the Unvoiced Consonants group, although each phone may be alternatively be modeled individually in the future. All other vowels are modeled by individual phone models, with the exception of certain first vowels of the Nulls group - those which are particularly abrupt are modeled like biphones as though there were unvoiced consonants in front of them.

For phone units, one wavetable is used per phone model, with approximately one model every three semitones across the vocal range. The wavetable is tuned and normalized. Its power envelope is flattened and zero-centred. Finally, its set to start and end at the same phase so they may be set to loop. Biphone units are conditioned in the same way, except two wavetables are used per phone model. The first captures the initial phone of the sequence as well as the transition into the next phone. The second is of the second phone. It is set such that the second wavetable starts at the same phase as the end of the both the first wavetable and itself. In this way, the second wavetable may be set to continue after the first wavetable and set to loop. Figure 1 shows examples of the wavetables for phone and biphone units, illustrating the aforementioned phase observations.

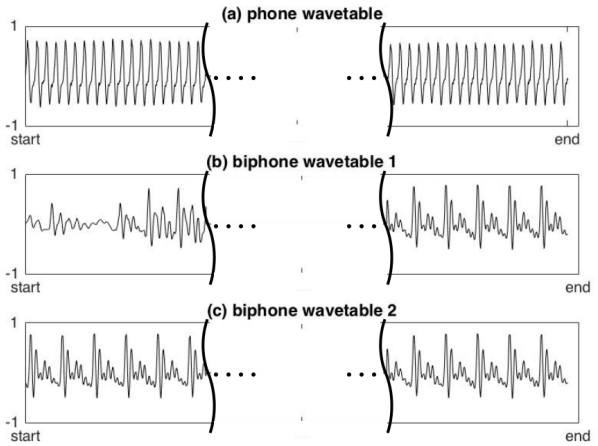

Figure 1: Phone- and Biphone-Model Wavetables

2.1.3. Syllable Model Database

For each class of syllables, for each possible final, a syllable model is built. The syllable model database consists of

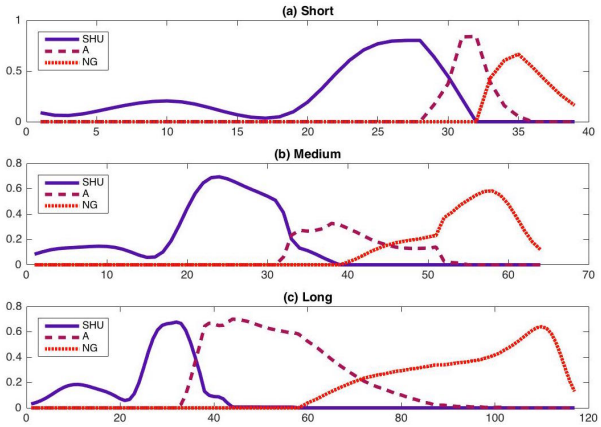

Figure 2: Wave Additive Trajectories across 3 Different Syllable Lengths for the Mandarin Syllable 'Shuang'.

wave additive trajectories. The syllable models indicate the contribution of each composite phone to the overall timbre at each time frame. Figure 2 illustrates the wavetable trajectories across three different syllable lengths for the Mandarin syllable 'Shuang'. Across all three plots, the solid blue line plots the trajectory of the first phone, 'SHU'; the dashed purple line plots the trajectory of the second phone, 'A'; and the dotted red line plots the trajectory of the last phone, 'NG'. At least four lengths per syllable are modeled in SERAPHIM, spanning a semiquaver to a semibreve for a tempo of 120 beats per minute. These are computed using labeled wave samples of the syllables. Wave samples to be modeled are first labeled to mark syllable and phone transition boundaries, as well as the centre of each phone. A DTW-like method is then used to traverse a matrix of their spectral similarities as illustrated in Figure 3. The transition lines are finally superimposed onto the power envelope to produce the wave additive trajectories. Wave additive trajectories are normalized in power and time and stacked to form a surface to form the syllable model which may be interpolated to invoke the model. An alternative method would be to replace the syllable model with a bell-like equation for better invoking speed but poorer detail.

2.2. Algorithm

2.2.1. Runtime Algorithm

The runtime system is illustrated in Figure 4. As shown in the figure, the lyrics $\Phi(t)$ at the input is first converted into a sequence of phone labels, $\phi(t)$. Given this, together with the melody, $f_0(t)$, the corresponding phone sample, $\sigma_\phi(f_d, \tau)$ and amplitude trajectory, $a_\phi(t)$ are retrieved from their respective databases. Since the phone sample's pitch may not completely match the specified pitch, $f_0(t)$, it then needs to be shifted to the correct pitch to $\sigma_\phi(f_d, t)$ before being tapered by coefficient $a_\phi(t)$ to produce a single frame segment of the output singing voice, $\sigma(t) = a_\phi(t)\sigma_\phi(f_d, t)$. (Since the shift is usually very small with adequate samples across the vocal range, it is efficient to do it by interpolation or resampling, and $\tau/t = f_0/f_d$.) These frames are finally concatenated to produce the output singing voice, $\Sigma(t) = [\sigma(1), \sigma(2), \sigma(3), ...]$.

2.2.2. Real-time Algorithm

The real-time version of the system is illustrated in Figure 5. As shown in the figure, input gestures are first interpreted to acquire phone labels, $\phi(t)$, melody, $f_0(t)$, and amplitude trajectories, $a_\phi(t)$. Given the phone label, $\phi(t)$, the corresponding phone sample, $\sigma_\phi(f_d, \tau)$ is retrieved from the phone model database. This is pitch-shifted to the instantaneous pitch input, $f_0(t)$, to produce $\sigma_\phi(f_d, t)$, which is further tapered by the input amplitude coefficient, $a_\phi(t)$, to produce the current frame of singing voice, $\sigma(t) = a_\phi(t)\sigma_\phi(f_d, t)$. Since this is a real-time algorithm, concatenation is inherent in the flowchart and not explicitly shown in the figure.

3. Implementation

The SERAPHIM engine finds numerous applications real-time and offstage alike. Current implementations are SERAPHIM *Live!*, ZECHARIAH, SERAPHIM Studio and SERAPHIM Toolbox. This section covers these real-time and offline implementations of the system and explains how different imple-

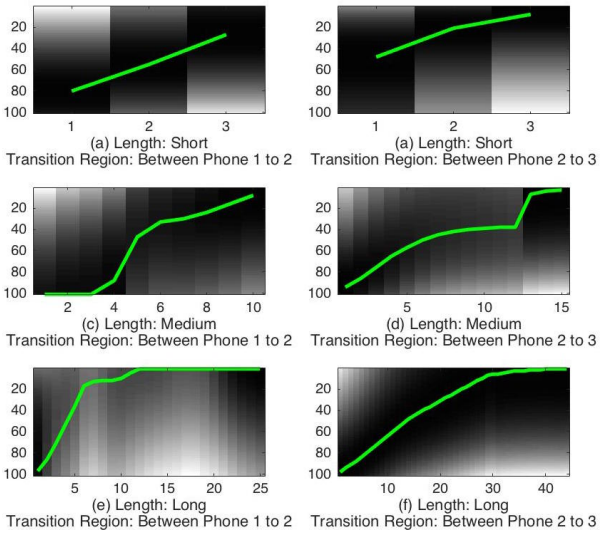

Figure 3: Computing the Transition between Phone Units

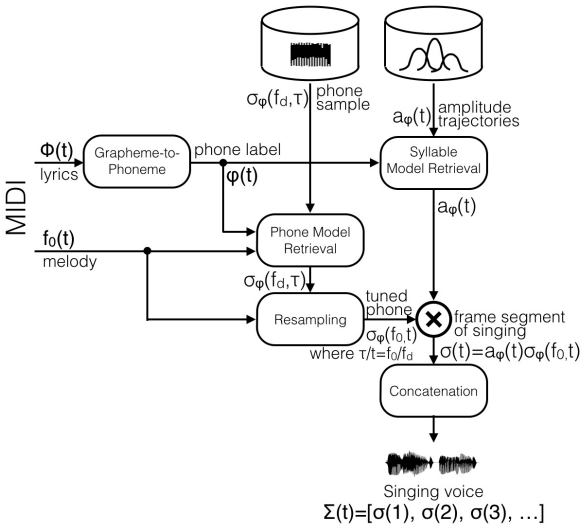

Figure 4: Runtime Synthesis Flowchart

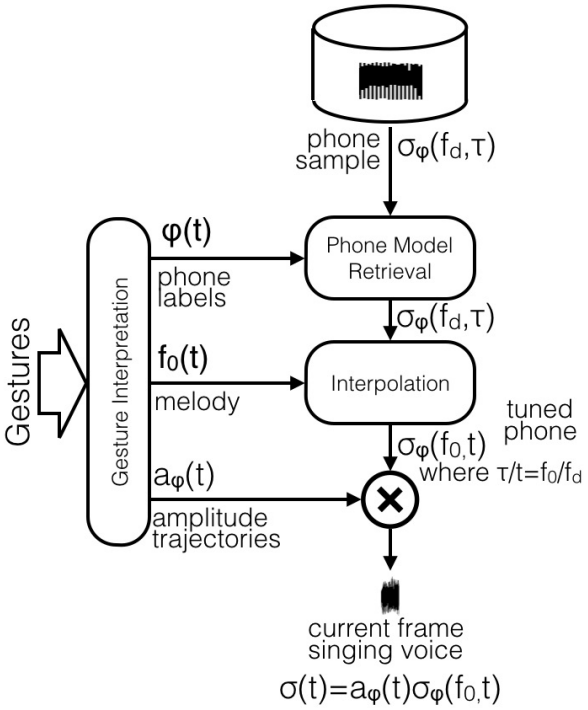

Figure 5: Real-time Synthesis Flowchart

mentations give performers, composers and fellow developers access to the features of SERAPHIM.

3.1. SERAPHIM *Live!*

SERAPHIM *Live!* is a musical-instrument styled real-time singing synthesis system that translates gesture inputs into real-time singing on portable platforms.

3.2. ZECHARIAH

ZECHARIAH is the real-time speech synthesis version of SERAPHIM *Live!* that provides a means for the mute to deliver real-time speech.

3.3. SERAPHIM Studio

SERAPHIM Studio is an implementation of SERAPHIM in Steinberg's VST technology, enabling the system to be controlled by a composer from within any music sequencer that supports VST technology. The demo submitted is a pre-release version of SERAPHIM Studio that also integrates Kio's model[19] of Hatsune Miku[2] to demonstrate how SERAPHIM's wave additive trajectories may be easily converted into signals to control a 3D character model for lip synchronization.

3.4. SERAPHIM Toolbox

SERAPHIM Toolbox is a Unity 3D package containing the SERAPHIM engine. This gives 3D game developers easy access to real-time singing synthesis in their games and provides real-time control signals for lip synchronization of 3D characters.

[2]Hatsune Miku, ©Crypton Future Media, Inc. 2007

4. Experiment

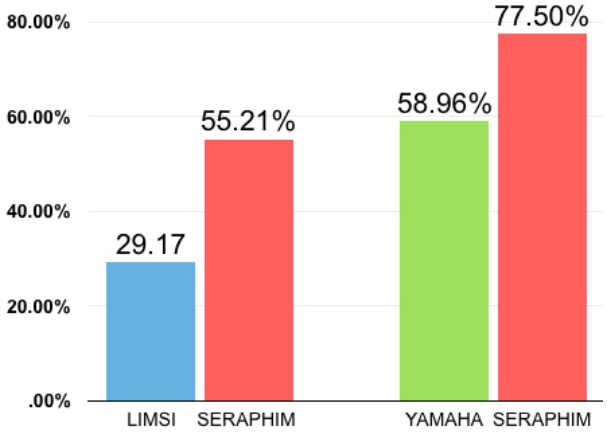

Figure 6: Subjective Scores of SERAPHIM against LIMSI's Cantor Digitalis and Yamaha's VOCALOID respectively

Four segments[20] of singing voices synthesized using LIMSI's Cantor Digitalis and four[21] using Yamaha's Vocaloid Keyboard were each mimicked using SERAPHIM *Live!*. The 16 segments in total were randomized and presented to 12 listeners in a subjective listening test, who were tasked to score the voices on a scale of 1 to 10, with 1 being the worst sounding and 10 being the best sounding. The results are normalized and presented in Figure 6. As shown in the figure, the SERAPHIM outperforms Cantor Digitalis by 26.04% and Vocaloid Keyboard by 18.54%.

5. Conclusion

Wavetable synthesis presents a suitable method of singing synthesis to be implemented live on a mobile platform. In this paper, we have introduced our system, SERAPHIM, a wavetable synthesis system for real-time singing applications on mobile platforms. We gave a comprehensive background of the system, explained its databases and algorithms, and presented some implementations of the system as well as subjective listening test results. From our results, it may be seen that our method outperforms each of the other 2 known real-time singing systems by more than 18%.

6. References

[1] J. O. i. Font, "Musical and phonetic controls in a singing voice synthesizer," Ph.D. dissertation, Polytechnics University of Valencia, 2001. [Online]. Available: files/publications/pfc2001-jortola.pdf

[2] S. Lemmetty, "Review of Speech Synthesis Technology," *Helsinki University of Technology*, vol. 320, pp. 79–90, 1999. [Online]. Available: http://www.acoustics.hut.fi/~slemmett/dippa/$\backslash$nhttp://www.acoustics.hut.fi/publications/files/theses/lemmetty_mst/

[3] L. K. Le, "Examining the Rise of Hatsune Miku: The First International Virtual Idol," *The UCI Undergraduate research Journal*, 2014.

[4] B. Larsson, "Music and Singing Synthesis Equipment (MUSSE)," *Dept. for Speech, Music and Hearing, Quarterly Progress and Status Report (STL-QPSR)*, vol. 18, no. 1, pp. 38–40, 1977.

[5] X. Rodet and X. Rodet, "Synthesis and processing of the singing voice," in *Proc.1st IEEE Benelux Workshop on Model based Processing and Coding of Audio (MPCA-2002)*, Leuven, Belgium, 2002, pp. 99–108.

[6] M. Clarke and X. Rodet, "Real-time FOF and FOG Synthesis in MSP and its Integration with PSOLA," in *Proceedings of the International Computer Music Conference*, 2003. [Online]. Available: http://nagasm.org/ASL/icmc2003/closed/CR1040.PDF

[7] L. Feugère, S. L. Beux, and C. D'Alessandro, "Chorus Digitalis Polyphonic Gestural Singing," in *INTERSPEECH 2011*, 2011.

[8] P. R. Cook, "Identification of Control Parameters in an Articulatory Vocal Tract Model, with Applications to the Synthesis of Singing," Ph.D. dissertation, Stanford University, 1991.

[9] M. W. Macom, L. Jensen-Link, J. Oliverio, and M. A. Clements, "A SInging Voice Synthesis System based on Sinusoidal Modeling," *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, pp. 435–438, 1997.

[10] J. Bonada, O. Celma, A. Loscos, J. Ortolà, and X. Serra, "Singing Voice Synthesis Combining Excitation plus Resonance and Sinusoidal plus Residual Models," in *Proceedings of International Computer Music Conference*, 2001.

[11] H. Kenmochi and H. Ohshita, "VOCALOID-commercial singing synthesizer based on sample concatenation." in *Interspeech*, no. August, 2007, pp. 3–4. [Online]. Available: http://www.mirlab.org/conference_papers/International_Conference/Eurospeech2007/NOREVIEW/PDF/AUTHOR/NORV1312.PDF

[12] H. Kenmochi, "Singing Synthesis as a New Musical Instrument," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 5385–5388.

[13] Ameya Purojekuto. Ameya/Shobu, "UTAU-Synth," http://utau-synth.com, 2008.

[14] K. Oura, A. Mase, Y. Tomohiko, S. Muto, Y. Nankaku, and K. Tokuda, "Recent Development of the HMM-based Singing Voice Synthesis System - Sinsy," in *7th ISCA Workshop on Speech Synthesis*, 2010, pp. 211–216. [Online]. Available: http://20.210-193-52.unknown.qala.com.sg/archive/ssw7/papers/ssw7_211.pdf

[15] K. Oura, A. Mase, Y. Nankaku, and K. Tokuda, "Pitch Adaptive Training for HMM-based Singing Voice Synthesis," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing*, 2012, pp. 5377–5380.

[16] J.-y. Cheng, Y.-c. Huang, and C.-h. Wu, "HMM-based Mandarin Singing Voice Synthesis Using Tailored Synthesis Units and Question Sets," *Computational Linguistics and Chinese Language Processing*, vol. 18, no. 4, pp. 63–80, 2013.

[17] C.-W. Wun and A. Horner, "Perceptual Wavetable Matching for Synthesis of Musical Instrument Tones," *JAES*, vol. 49, no. 4, pp. 250–262.

[18] C. Roads, *The Computer Music Tutorial*. MIT Press, 1996.

[19] Kio, "Blender," 2008. [Online]. Available: http://kiomodel3.sblo.jp/article/23486819.html

[20] Audio Acoustique LIMSI CNRS, "Cantor Digitalis - JDEV 2013, Palaiseau - Cantate 2.0," 2013. [Online]. Available: https://www.youtube.com/watch?v=d4TV-IcK8c8

[21] DigInfo TV, "Yamaha Vocaloid Keyboard - Play Miku Songs Live! #DigInfo," 2012. [Online]. Available: https://www.youtube.com/watch?v=d9e87KLMrng