KNUTH AND ALMA

TWO PARTNERS FOR LIVE-ELECTRONICS WITH SPOKEN WORD

Joachim Heintz
joachim.heintz AT hmtm-hannover.de

The main idea behind *Knuth and Alma*[2] is the development of a simple live-electronic setup for a speaker. It follows the german child song "Ich geh mit meiner Laterne", which describes a couple of a child and a lantern. For the speaker, their lantern is a small loudspeaker; small enough to be carried and to be put on the table, beneath the speaker when they read a text, large enough to be a counterpart to the human voice.
Although the setup is the same for both, Knuth and Alma focus on quite different aspects of spoken language, and output quite different sounds. Knuth analyzes the rhythm of the language and triggers pre-recorded samples of any sound, whereas Alma recalls parts of the speaker's past and does not use any other sound except the speaker's live input itself.
I will first describe the two models and their implementation in Csound, and then I will discuss some use cases and future possibilities.

KNUTH

Rhythm Analysis

The most significant rhythmical element of speech is the syllable. The poem's metric counts syllables and distinguishes marked and unmarked, long and short syllables. Drummers in many cultures learn their rhythms and where to beat the drum by speaking syllables together with drumming. Vinko Globokar took this practice in his piece *Toucher*, thereby musicalizing parts of the *Galileo* piece of Bertolt Brecht.
But how can we analyze the rhythm of spoken words? Let us look at the famous community example, the "quick brown fox" from the Csound Manual.[3]

[2] Why names? Because they are individuals. – Why these names? Because they fit. No relation to the names of living or dead persons is implied.

[3] http://csound.github.io/docs/manual/examples/fox.wav

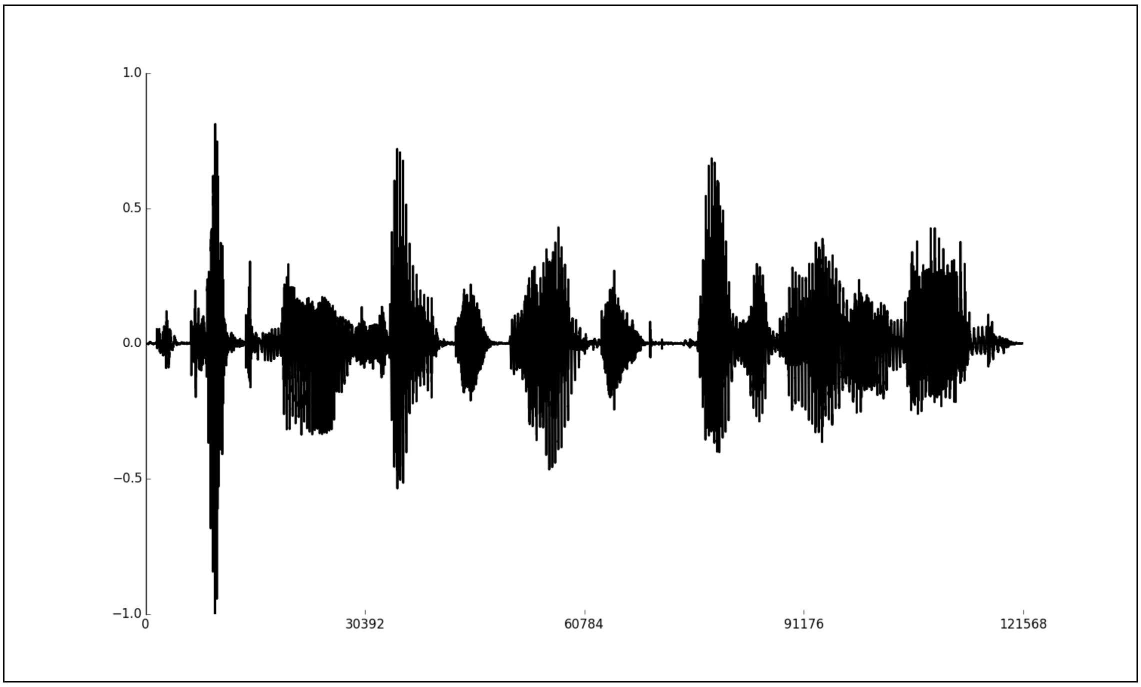


Figure 1 "the quick brown fox jumps over the lazy dog"

What we would like to get out is what our perception does: a signification — for instance an impulse — shortly after a vowel can be recognized. Like this:

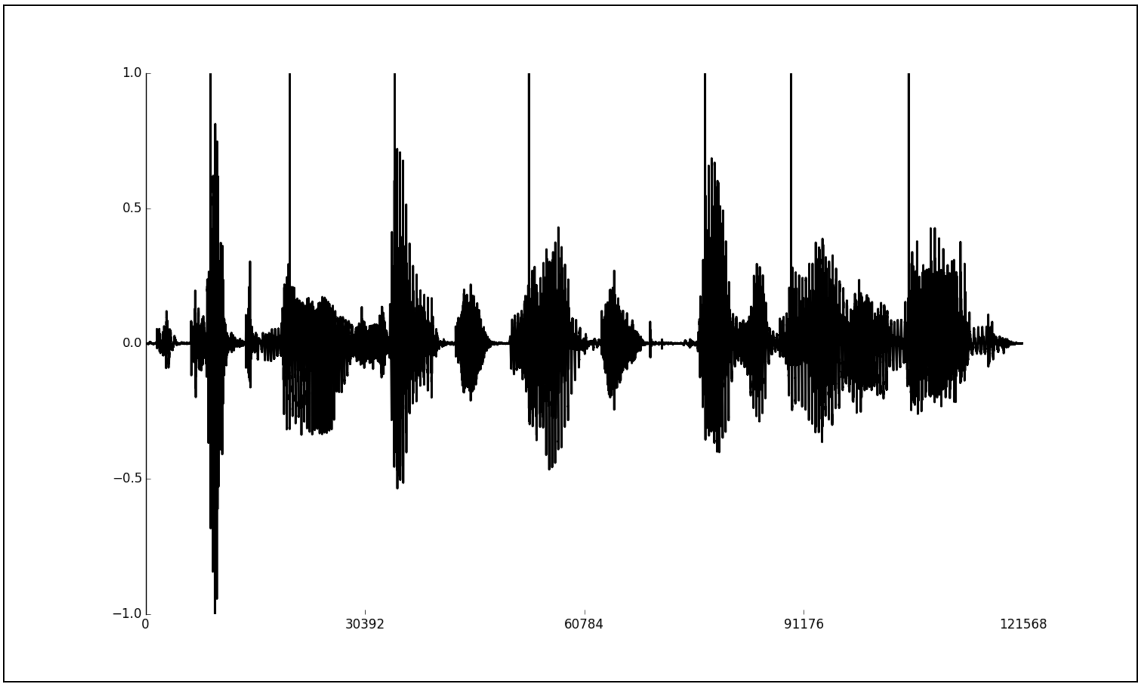


Figure 2 Desired recognition of syllables: the quIck brOwn fOx jUmps Over the lAzy dOg

After some not very successful trials with onset detection in the time-domain, I came to good results using FFT. The Csound opcode *pvspitch* attempts to analyze the fundamental of a signal in the frequency domain.[4] Considering that a vowel is the

[4] The author of the opcode wrote an excellent article in the Csound Journal: Alan O. Cinneide, Introducing PVSPITCH, A Pitch Tracking Opcode for Csound, Csound Journal, Issue 2, Winter 2006. (http://csoundjournal.com/2006winter/pvspitch.html)

harmonic part of speech, this should coincide with what is the task here. A threshold *gkThreshDb* can be assigned, to exclude everything which is certainly no peak because it is too soft. If no fundamental can be analyzed, *pvspitch* returns zero as frequency. So looking for the zero-to-non-zero transitions, we can get a first version of the speech rhythm detection:

```
/*initialize the previous state of frequency analysis to zero hz*/
kFreqPrev init 0

/*set FFT size*/
ifftsize = 512

/*perform FFT*/
fIn pvsanal gaLiveIn, ifftsize, ifftsize/4, ifftsize, 1

/*analyze input*/
kFreq, kAmp pvspitch fIn, ampdb(gkThreshDb)

/*ask for the new value being the first one jumping over kthresh*/
if kFreqPrev == 0 && kFreq > 0 then

    /*trigger subinstrument*/
    event “i”, “whatever”, 0, 1

endif

/*update next previous freq to this freq*/
kFreqPrev = kFreq
```

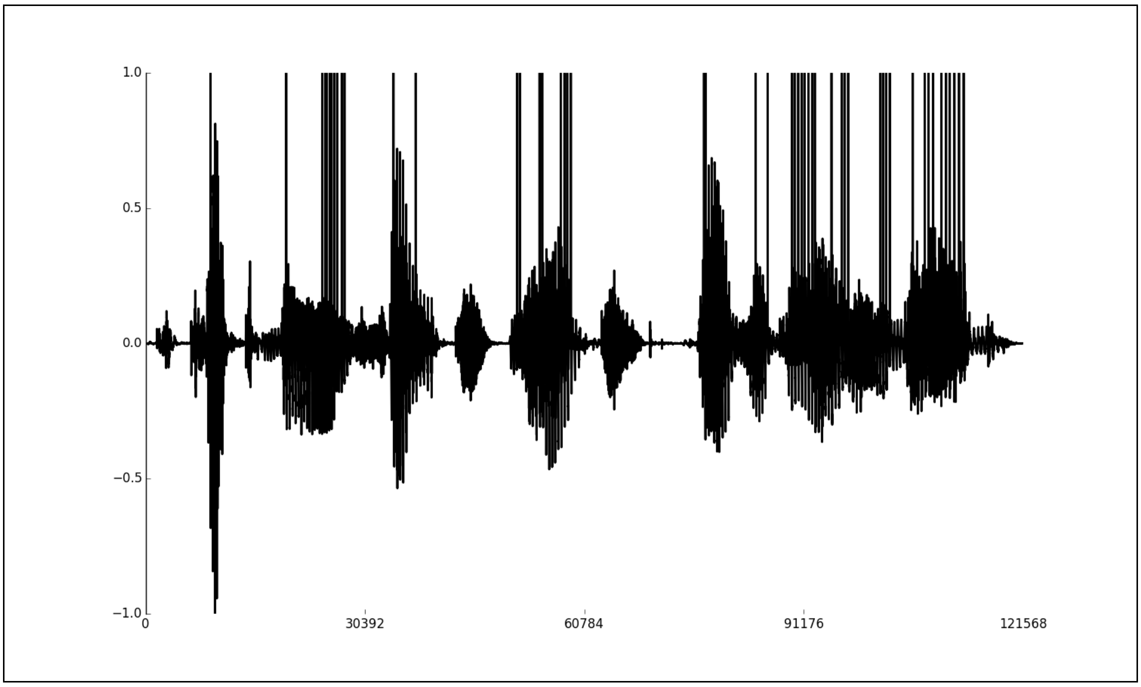


Figure 3 Recognitions without time limit

Actually, all we have to do now is to get rid of the repetitions. This can be done by setting a minimum (refractory) time before the next analysis can be performed. This is the result, with 0.2 seconds of minimal time interval between two analyses:

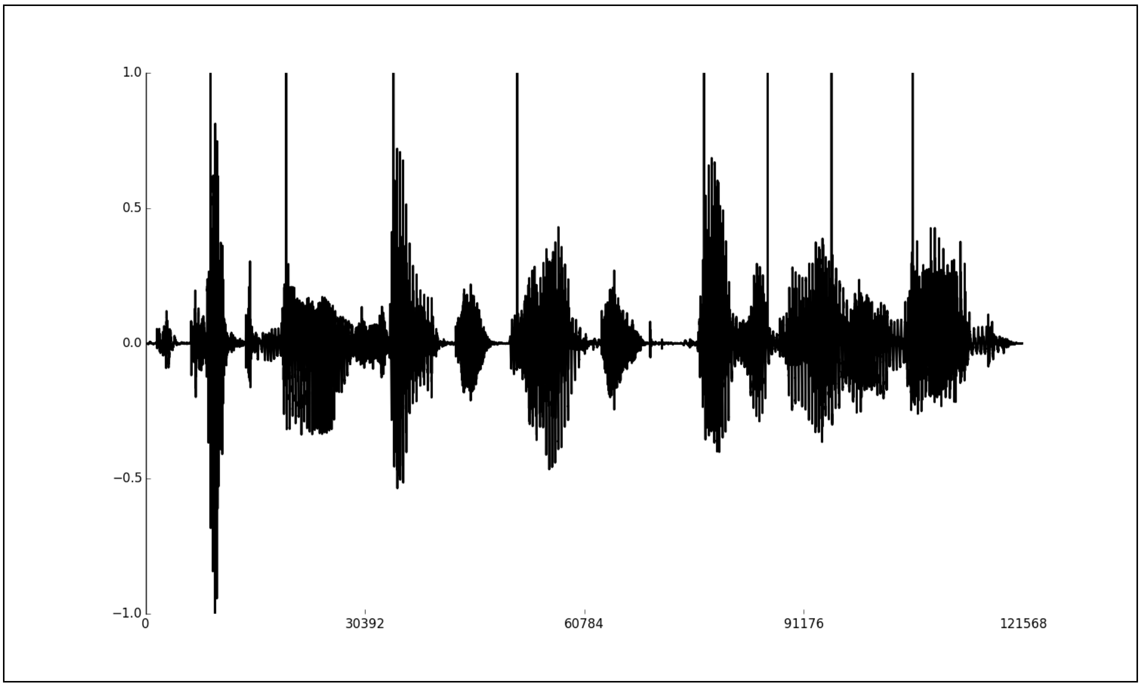
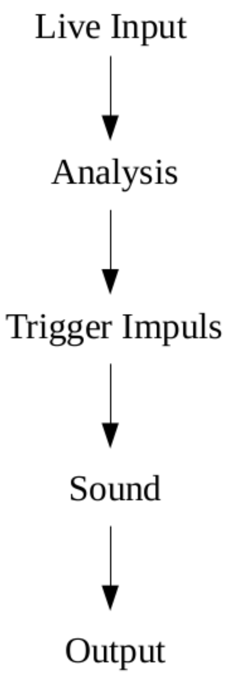


Figure 4 Recognitions with time limit = 0.2 seconds

General Setup

The basic idea of Knuth is to trigger a percussive sound whenever a new syllable is detected. So the main flow is simple:



But there are many very different variants and possibilities in the single parts of this chain. First, "Sound" consists of different units:

– Which sound? Sampled or synthesized sound? — Currently I am only using sampled sound. These sounds usually have a strong atmosphere; a group of them can build a homogenous or heterogeneous space. Should there be one sound at a time, or a mixture of two, three, or more? If a mixture, in which proportions?

– Which modification? I use mainly transposition and reverb. As the pvspitch analysis returns the frequency of the estimated fundamental, the transposition can be applied straightforward, depending on this frequency. Reverb is used in conjunction with the transposition, so that lower sounds have more reverb.

– Which mode of playing? The simplest way is to play out directly, in real time, simultaneously with the live input. This is nice, and in a way the proof of concept, but it has turned out that a delayed playback offers many new possibilities. Another mode could be to collect a certain amount of impulses, and play them as a group, or to play a sound only if condition X is given.

Given this, the next question arises: How should the decision between the choices be made? Assumed we want to choose between three sounds, A, B and C, and we get the trigger impulse with the information that the fundamental is 288 Hz and the amplitude –7 dB. Some possibilities of choosing would be:

– Choose by frequency. If we have mapped a region for frequencies below 200 Hz to sound A, a region 200...400 Hz to sound B, and a region above 400 Hz to sound C, sound B will be chosen here.

– Choose by amplitude. Similar, if amplitude below –20 dB is mapped to sound A, –20.. –10 dB to sound B, and –10...0 dB to sound C, sound C will be chosen here.

– Choose randomly. The probability of the sounds can in this case be either equal or not, for instance we could give sound A a probability of 0.6, sound B of 0.3, and sound C of 0.1, so that the latter will be very rare. And of course, all the well-known random walks can be applied here.

So actually, what at the first view looked very simple, opens up at a closer view a variety of possibilities and decisions — decisions which are mainly of artistic nature, and which will have a major impact on the musical result.

Implementation

The current analysis has already been discussed above. Other methods are possible, like cepstrum or band filter responses.[5] The main work in programming was to find a structure which has a maximum possible flexibility for changes, for further development and adaptions (to a special performer, a situation, a new idea). I ended up with this:

– An “always on” instrument receives the live input signal and the GUI input. It starts all subinstruments.
– An Analysis instrument performs the analysis, according to settings which are mainly received by the GUI.

[5] Perhaps even pure onset detection works better than I found out in my first tests.

– Once a new syllable has been detected, the analysis instrument calls one or more instruments which apply the results in selecting the sampled sound(s), transposition, reverb and delay.

– Finally, the sampled sound is played out by another instrument.

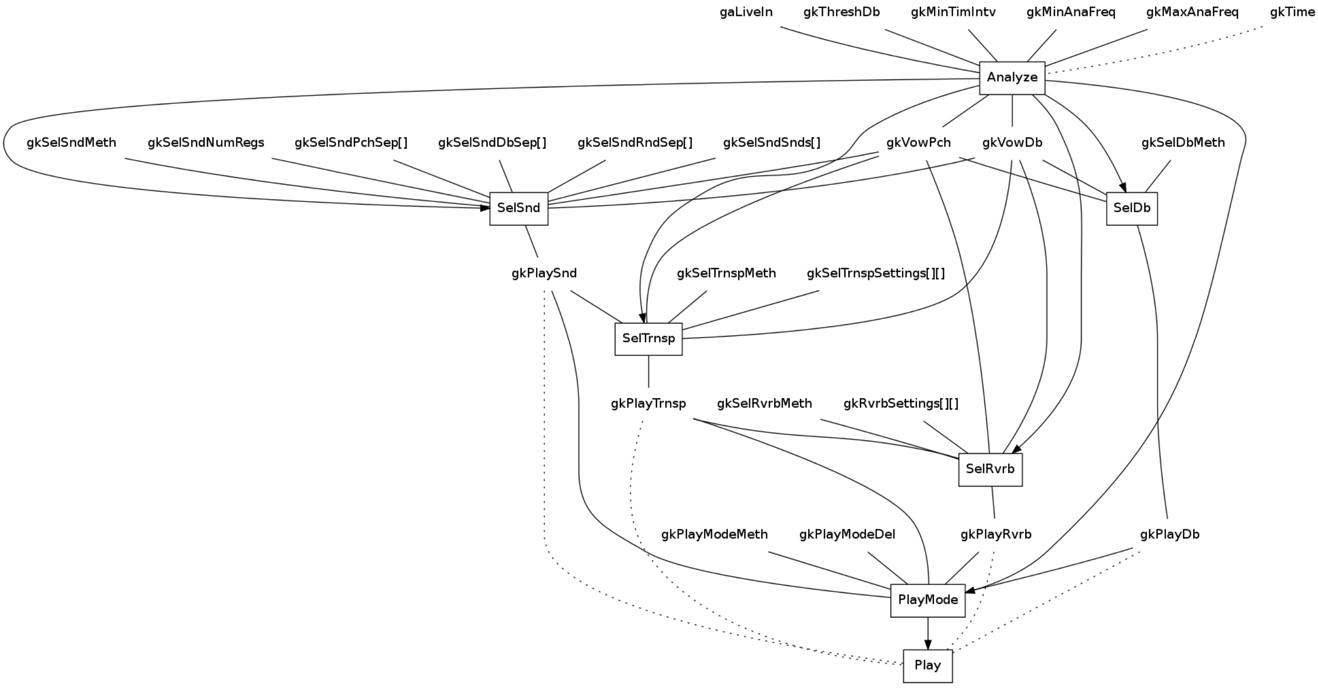


Figure 5 Knuth's control flow

Alma

Game of Times

Imagine someone who is reading a text. While he is reading, parts of what he already read are coming back, in certain forms or modes. Parts of the past are coming back, thus confusing the perception of time as flow, as succession. A Game of Times begins, and the text changes its face. Instead of a stream, always proceeding from past to future, it becomes a space in which all that has gone can come back and be here, in this moment, in the present. A line becomes a collection of fragments, and the fragments build up a new room, in which no direction is preferred. You can go back, you can go ahead, you can cease to move, you can jump, you can break down, and you can rise again in a new mode of movement. But definitely, the common suggestion of a text as succession will experience strong irritations.

Speech as Different Sizes of Sounding Matter

So Alma is about the past, and it only works with the material the speaker has already uttered. But this material is not equivalent to all that has been recorded in a buffer. It would be unsatisfactory to play back some part of this past randomly: sometimes a syllable, sometimes the second half of a word, sometimes silence.

No, the sounding matter must be analyzed and selected. The decision for Alma is this: Start by recognizing sounding units of different sizes. A very small size of such a sounding unit matches approximately the phonemes; a middle size matches the syllables; a larger size matches whole words or even parts of sentences.

The number of sizes or levels is not restricted to three; there can be as many as you like, as many as you need for the game of times. The method used to distinguish a unit is very easy. A sounding unit is considered as something which has a pause before and afterwards. The measurement is simply done by rms[6]; the question is only about the time interval over which the rms value is measured. A shorter time interval for rms estimation will isolate smaller units; a longer time interval will separate larger units. Let us see this exemplified by the “quick brown fox” again. The figure shows three minimum time intervals and the results as analyzed sounding units.

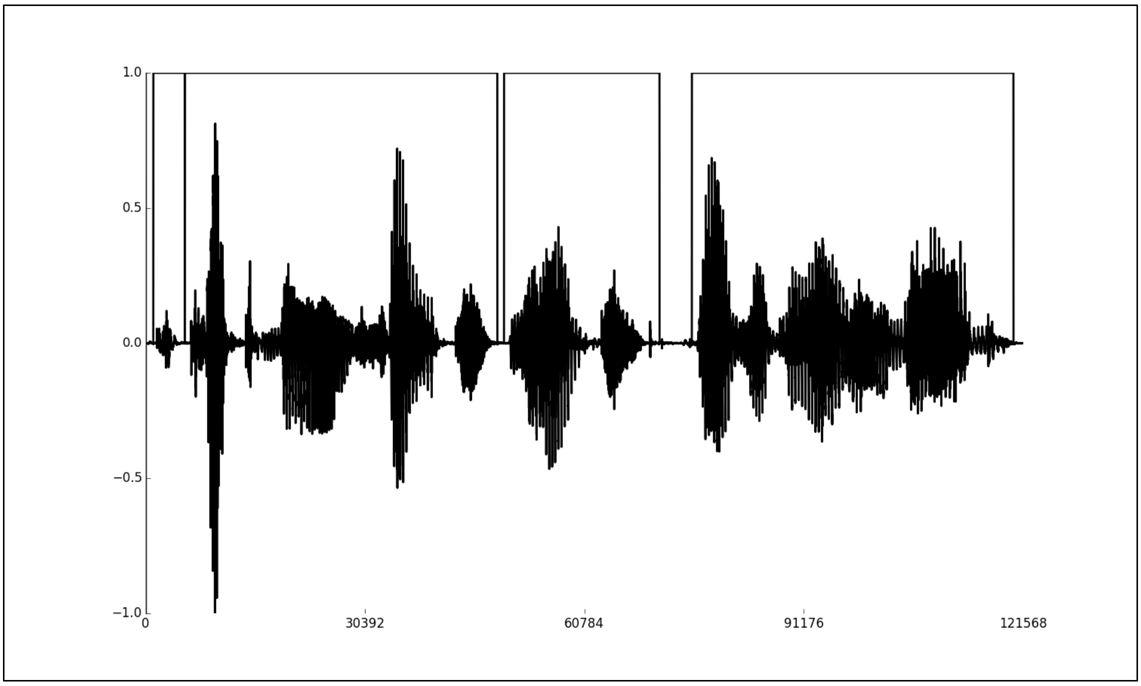


Figure 6 Analyzed units with minimum silence time = 0.04 seconds

[6] the root of the arithmetic mean of the squares of a number of samples (= amplitudes), using Csound's rms opcode (http://csound.github.io/docs/manual/rms.html)

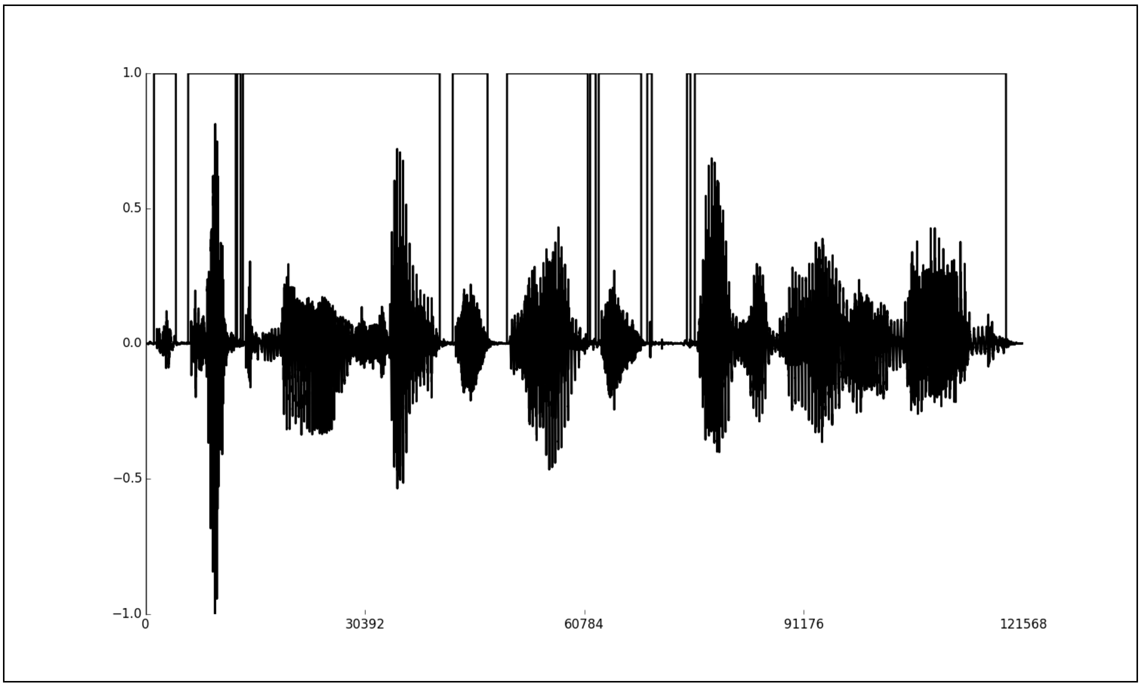


Figure 7 Analyzed units with minimum silence time = 0.01 seconds

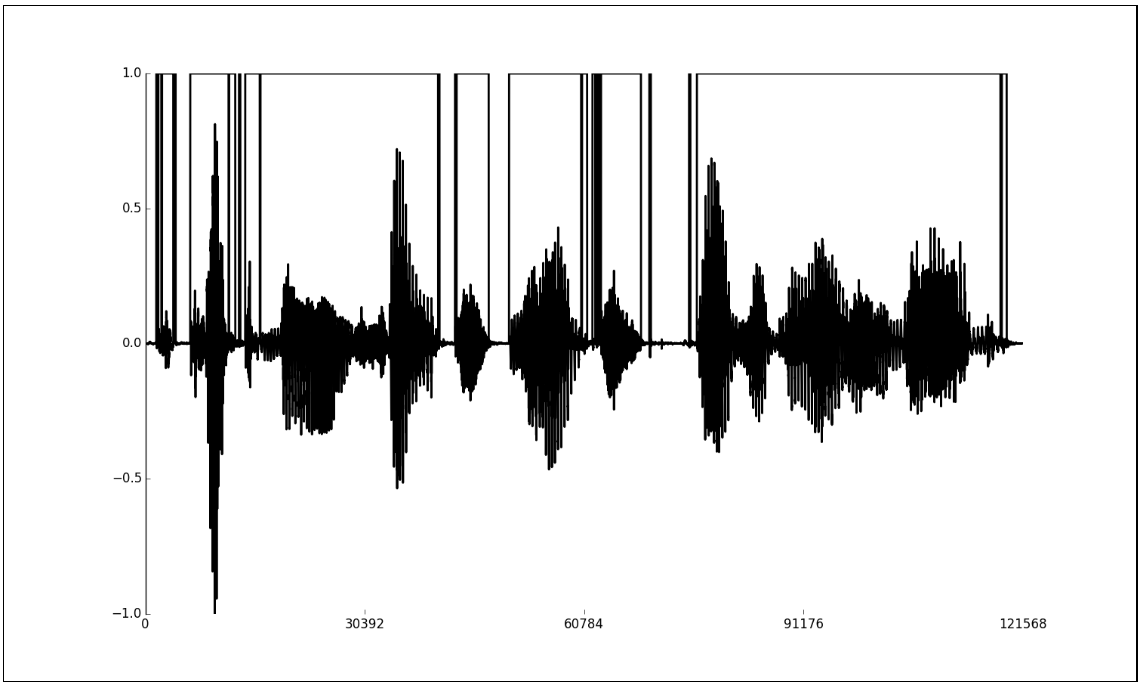


Figure 8 Analyzed units with minimum silence time = 0.002 seconds

As can be seen, although the analyzed units become smaller and more numerous, some units may nevertheless remain very large, is the speaker avoids isolating but speaks more like a singer.[7]

[7] It is easy to insert a maximum time for the units in the code, if this is desired.

The result depends also very much on the threshold below which "silence" is considered to be. In the figures above, the threshold is –40 dB. If set to –20 dB instead, the first result changes to this:

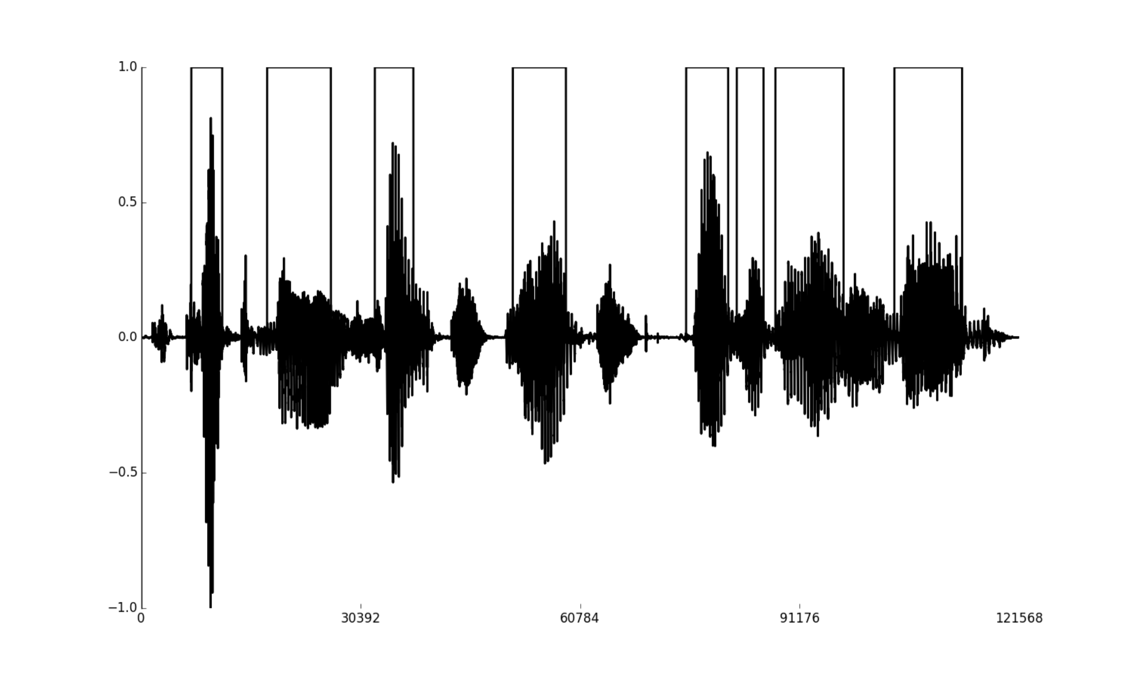


Figure 9 Analyzed units with minimum silence time = 0.04 sec and threshold = –20 dB

It is worth to note that the results do only roughly correspond with the above mentioned distinction between phonemes, syllables and words. These latter ones are analyzed by meanings, but Alma deals only with the spoken language in one particular aspect. It very much depends on the speaker, her way of connecting and seperating the sounds, weather or not "usual" units are analyzed by Alma, or unexpected ones are derived.

Bringing Back the Past: Four Modes

Currently Alma can bring back the past in four different modes. None of these modes is a real "play back". All change or re-arrange the sounding past in a certain way.
For all modes, the speaker-performer can choose between four different selections of the past. This is a bit different for the single modes, but in general the first selection grasps the immediate past, the second selection points to a region some seconds ago, the third selection covers a region more wide and further in the past, and the fourth selection takes a sound snippet from all which has been recorded since the beginning of the performance.

This is an overview of the modes:

Wave. A large number of small speech particles create a sound which resembles a wave breaking at the seashore. This is done with a special way of scratching, combined with a variable delay-feedback unit.

New Language. Units in the overall size of syllables are put together in a new order, so that new words are created. Pauses between the syllables can be added, so that instead of words a more scattered image is created, so to say a landscape of syllables.

Rhythm. A short rhythm is given in proportions, for instance 1/2, 2/3, 1/3, 1/4, 3/4, 1. This rhythm controls the playback of isolated sound units, so that the natural, free rhythm of the language is left in favor of a rhythm in measurements. To avoid pure repetitions, the rhythm is varied in applying permutations and different scalings of expansion/compression.

Flight. A sound snippet is transformed into a bell-like sound which gently seems to speak. This sound can be of very different durations, starting from the original (= short) duration, until a stretch factor of thousand. Although it reproduces the most prominent partials of the original sound, it sounds high and adds a pitched, slowly decaying sound to the overall image.

Some Implementation Details

All incoming sound is written into a buffer (function table). An array consisting of pairs of (a) minimal silence time for setting a marker, and (b) the maximum number of markers is read by the program at initialization. As many function tables of the desired length are created as pairs are written. This array, for example, will create four function tables:

```
    [.005, 100000, .02, 10000, .01, 50000, .05, 10000]
```

The first table of length 100,000 will be used to write markers after a silence of 5 milliseconds or longer. The second table of length 10,000 is for markers after a silence of minimum 20 milliseconds, and so forth.

The instrument *WriteMarker* analyzes the rms and writes markers in a function table. There are as many instances of this instrument as there are tables to write in. As the tables are numbered 1, 2, ... the communication between an instance and its table can be done via software channels with dynamically created names:

```
;get values
iTableNum = p4
iTableLen = p5
iSilMinTim = p6

;set channel for passing the marker number to other instruments
S_MarkerChnl sprintf "MaxMarker_%d", iTableNum
chnset kMarkerNum, S_MarkerChnl

;set channel for passing the minimum silence time to other instruments
S_SilTimChnl sprintf "SilMinTim_%d", iTableNum
chnset iSilMinTim, S_SilTimChnl
```

The four instruments (*Wave, NewLang, Rtm, Flight*) receive the current marker number, defining the maximum possible marker to read from. As WriteMarker writes a marker each time "silence" starts or ends,[8] the receiving instrument knows where a sounding unit begins, and where it ends.[9]

[8] Technically spoken: each time the rms value crosses the dB threshold after a certain duration, either from above to below the threshold, or vice versa.

[9] Markers count from zero, but the changed() expression leads to a first increment when the program starts. So odd markers define the start of a sounding unit, even markers define its end. The rms analysis duration can be subtracted, or used for a crossfade.

The four instruments are triggered via a MIDI Keyboard. Each instrument can select one of four marker regions, as described above, so a two-octave keyboard is sufficient for the sixteen keys needed. Eight knobs on the same keyboard are used to control some parameters in real-time, like threshold level, duration for the *Wave* instrument, duration of additional pauses for the *NewLanguage* instrument, and the volumes for each of the four instruments.

I used CsoundQt as frontend, but it should be easy to adapt this program to Cabbage or any API application.

Performing with Alma and Knuth

General Setup

Alma and Knuth are waiting for performers, readers, storytellers to play with them. I have worked with different people in different setups, and would like to report here some experiences I have made until now.

Alma, the game with the past, seems to have two basic setups. The first is for one person alone. This performer learns to play with Alma, and triggers everything by their own. For this, a touchable interface much bigger and much nicer than a MIDI Keyboard would be ideal.

The second setup for Alma is for two persons: one speaker/performer, and one live-electronic player. In this case, the dialogue between live speaker and Alma is in a way transformed into a dialogue between two people. The speaker may be confronted with reactions of Alma, they may not have foreseen.

The general distinction between these two setups is valid for Knuth, too. Of course, Alma and Knuth can be combined easily, but not always “more” is “more”...

The performance can be anywhere between pure improvisation and fixed composition. It is a challenge for me to find ways of writing a score for this kind of text-based composition.

Speaker and Space

The main idea behind both, Alma and Knuth, is: one human speaker — one loud speaker. Usually the human speaker will not be amplified, so the loud speaker can be rather small. I got good results with a Yamaha MSP-5, or even MSP-3 studio monitor. Another variant would be completely mobile: Csound on Android, and a battery-driven speaker which can be carried. This would allow extending the performance to mobile man-speaker units.[10]

[10] Of course, a microphone is needed for all possibilities. I used a Shure SM58. For the mobile solution, a headset should work best.

Alma and Knuth need, I think, some intimacy, and a good contact with the audience. So in concordance with the small speaker size, a small space should be most appropriate. The Carnegie Hall will be worse than a workshop room.

Errors and Irritations

In a way it is part of the concept that Alma and Knuth are not perfect. Spoken language is not the same as the analysis of the spoken language we all do “on the fly” when we listen to someone who speaks. Alma will stick strictly to the sounding result; a cough is for her the same as a holy word. And Knuth may not recognize a syllable which you are sure you mentioned — be it because you did not pronounce it properly, or because the threshold was not chosen well.

As long as these irritations do not exceed a certain limit, they should motivate the speaker-performer to react to the unexpected, and to give it a new meaning.

Further Developments

Future work will very much depend on the people who play with Knuth and Alma, and their situations of usage. By intention these are concepts, not pieces. Their strength lies in their flexibility. They can be modified and they can learn, but they will always remain a variation on rhythm (Knuth), and a variation on the sounding past (Alma) of spoken language.

New needs arise while working with them, and I would love to understand these needs as experiments on how we can “read” texts — to open up our ears, to open up new ways of “understanding” a text ... — one of the most common and most strange things men ever created.

Acknowledgements

Thanks to Laureline Koenig and Tom Schröpfer for partnership in first developments of Alma and Knuth. Thanks to Anna for reading this text.