American Journal of Modeling and Optimization, 2016, Vol. 4, No. 2, 29-39
Available online at http://pubs.sciepub.com/ajmo/4/2/1

DOI:10.12691/ajmo-4-2-1

The Phonemes Recognition through Formant Analysis in Vowel-consonant Transition Case in "Baoule" Language of Côte d'Ivoire

H. KONAN[1], O. ASSEU[1,2,*], E. SORO[1], B. T. GOORE[2]

[1]Ecole Supérieure Africaine des Technologies d'Information et de Communication (ESATIC), Abidjan, Côte d'Ivoire
[2]Institut National Polytechnique Félix Houphouët Boigny (INP-HB), Yamoussoukro, Côte d'Ivoire
*Corresponding author: oasseu@yahoo.fr, hyacinthekonankouassi@yahoo.fr.

Abstract In this article, we are presenting the work we did in the context of the recognition of "Baoule" Language spoken phrases. Since we consider a particular case of the problem, we therefore will only take into account the vowel-consonant transition, while stopping on the Phoneme recognition. We are proposing a Phoneme separation algorithm, based on the consonants and vowels energy levels difference. We are also proposing a recognition algorithm by analysing the formants.

Keywords: *FFT(Fast Fourier Transform), spectograms, formants, phomemes, vocal trapeze, convolution, Harmin Window*

Cite This Article: H. KONAN, O. ASSEU, E. SORO, and B. T. GOORE, "The Phonemes Recognition through Formant Analysis in Vowel-consonant Transition Case in 'Baoule' Language of Côte d'Ivoire." *American Journal of Modeling and Optimization*, vol. 4, no. 2 (2016): 29-39. doi: 10.12691/ajmo-4-2-1.

1. Introduction

The machine automatic speech recognition, has long been a research topic that fascinates the public, but remains a challenge for specialists, and continues since then to be at the heart of scores of research. The new Information and communications Technology progress has allowed this research acceleration. Researchers have conducted electronics corpuses in developped countries [1]. Vast corpuses of tagged electronic texts are mainly available in English, French or German today. This has allowed the considerable development of automatic processingthese languages have known. Speech recognition systems using these corpuses are gradually expanding, while their Arabic [2], Turkish and vietnamese [3] equivalents are on the eave to emerge. There are languages spoken by most of the Fulani people as people do for the "Baoule" language in Cote d'Ivoire. Yet, there is no corpus to date for those languages, more, though the "Baoule" language uses French phonems, this language is not entirely covered by the latter. Indeed, acoustic units. Such as /bg/ or, /kp/ have no equivalent in french. We therefore cannot use French language artificial voices to recognize "Baoule" phonemes [4]. The inexistence of these"Baoule" phonemes, brought us to a collaborative research project with the ILA(Applied Language Institute) of the F. H. B. University, whose object is the construction of a reference corpus in "Baoule" Language. Once this corpus is available, we could engage in various research work in RAP.

We are proposing a Phoneme separation algorithm, based on the consonants and vowels energy levels difference, and we also propose a recognition Algorithm through analysis of the formants.

2. The Context

2.1. Here are Few Phonetics and Phonology Elements

2.1.1. The Sound and the Human Hearing

The sound is a wave propagating in a material medium, in the form of small variations pressure. It is perceptible to the human hear. when it's frequency is generally between 20 Hz, 20KHz or less. However, it is estimated that the phonetic information is below 10 KHZ. In the case of a telephone call, frequency above 3. 5 KHZ are cut, but the conversation remains intelligible, while some very acute phonemes, like/S/ are being badly rendered.

It is usual in signal processing to only consider spectra of amplitudes and shift phases spectra. This is due to the fact that when filtering audio signals, we simply change the amplitude spectrum because the ear is little sensitive to phase distortions [8]. this justifies that we only be interested in the following applications module.

2.1.2. The Human Voice

The human voice is the result of the on the way breath and the various phonatory organs interaction. In the voiced phonems case, the sound is initially produced by the vocal cords vibration. According to the cavities through which it passes, it is processed differently, pharynx and mouth mainly (Figure 1).

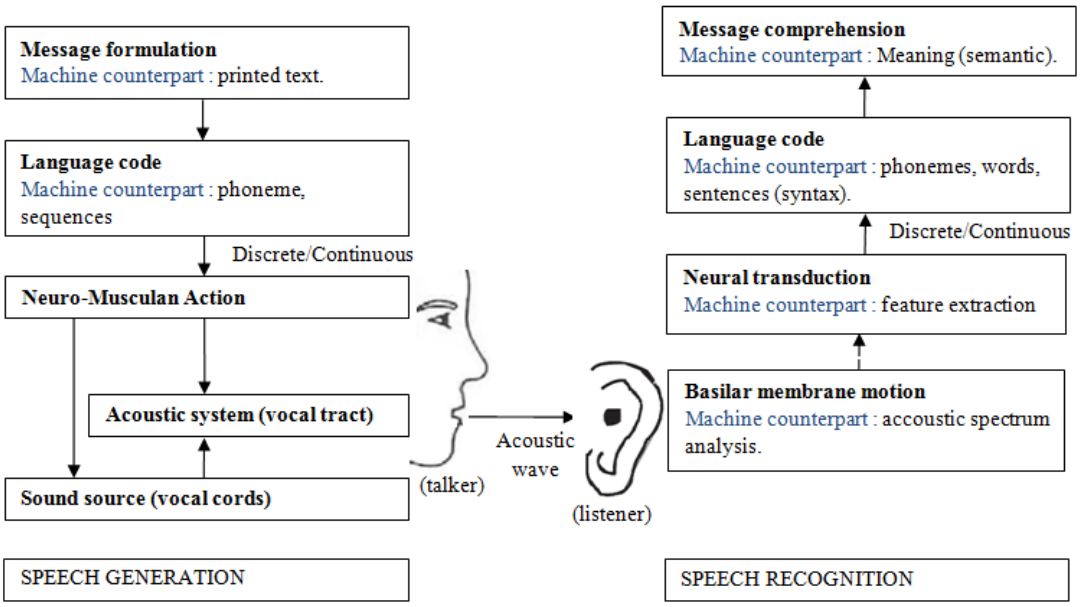


Figure 1. Production processus shematic Diagramspeech perception

For the frequencies corresponding to their resonant frequency, these cavities act as resonators reinforcing certain phonemes. These frequencies are called strengthened formants, and they are the one phonologists try to locate in spectogram. in order to recognize spoken phonemes. It is interesting to note that the nasal cavities, attenuates the formant said nasalised phonemes, and this is the reason why they are considered anti resonance phenomenon. These phonation organs are not frozen, their shape varies depending upon a particular Phoneme own hinge and their resonance frequencies, hence the different formant appearance for each Phoneme. In practice, we take into account the first three or four formants, being the most related to the articulation mode. In this research, we have decided to retain four, denoted F1, F2, F3 F4.

2.1.3. Phonetics and Phonology

Phonetics is the study of phonemes, the latter often defined as the "smallest distinctive sound unit". For example, the opposition of word bath and bread shows that /b/ and /p/ are phonemes. They can be classified into classes and subclasses, the first level being the vowels and consonants. The vowels are sounds that have a fairly long period. In the absence of prosodic elements too marked, the formants are horizontal on a spectrogram. Vowels are classified using the voweltrapezium [7], reproduced below (Figure 2).

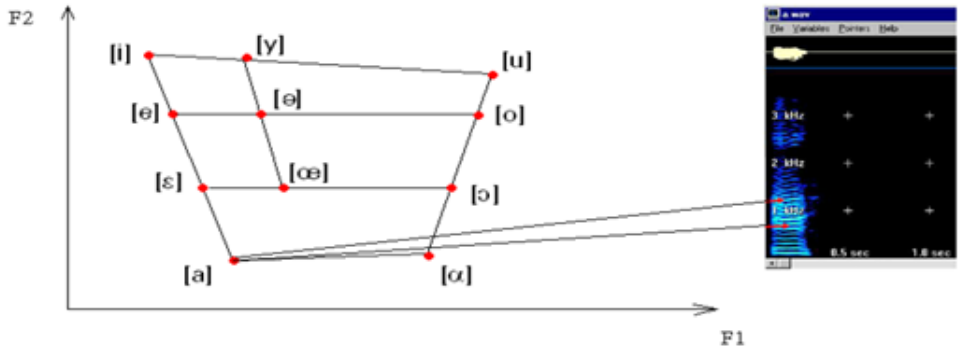


Figure 2. classification of vowels using the trapezium vowel

The consonants are phonemes which encounter an obstacle(for labial lips vowels teeth for dental, palace closure for [K] etc.)at their articulation. They are generally much shorter than the consonants and, much more variable overtime. They can be noisy or sonorants. It is only on this last case that they show formants.

Figure 3 shows the spectogram of the syllable[ASA] and illustrates some of the previously formulated concepts. We clearly can perceive. The horizontality of the vowel' s formants (even they are difficult to be distinguished from other frequency bands " parasites"), on this unprocessed image. The [s] consonant is a very sharp noise located above 5KHz, clearly free of formants.

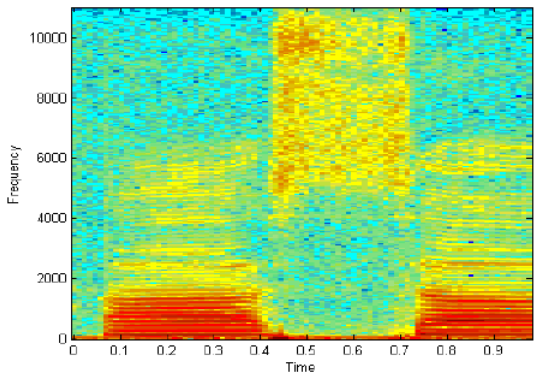


Figure 3. Syllable Spectrogramme [asa]

2.2. The "Baoule" Language

The "Baoule" language is spoken by about three million people in Côte d'Ivoire (source: Dictionary Baoulé-French) [4]. The "Baoule" lie amidst a vast area of 35, 000 km^2. The language "Baoule"from a genetic point of view, belongs to the "kwa" branch of the Niger-Congo family, one of four lingustiques families sharing the languages spoken on the African continent. In the subgroup "Bia", the "Agni" and the "Baoule" appear so close that we have not hesitated to speak of a single lingustique field, the "Agni-Baule" (CRESSEILS D. and N. KOUADIO . . 1997) a lexicostatistics study on languages "kwa" Côte d'Ivoire (BOLE RICHARD R. and LAPHAGE ph, 1983) shows an almost equal distance between "Baoule", "agni" and "Nzema": 68 % of correspondence between "agni" and "Baoule", 66. 9% from "agni" and "Nzema", 57. 8% among "Baoule" and "Nzema. "

2.2.1. The spelling of "Baoule"

The spelling of the "Baoule" presented here is the fruit of several years of research by both researchers at the Institute for Applied Linguistics (ILA) by independent researchers. The conventions used here, however, follow the principles established by the ILA under the spell standardization of Ivorian languages. All questions are not resolved, far from it, but we think we can say that even if changes are perfectly feasible on a particular point, they will not fundamentally challenge all of the following.

2.2.2. Inventory of Sounds (Phonemes)

The "Baoule" language can be described in terms of a set of about 33 distinct sounds called phonemes, illustrated in Table 1 and Table 2.

Table 1. "Baoulé" language vowels

API/ A. P. A.	Orthographe baoulé	Exemples
a	a	sa : main
e	e	je : dent
ɛ	ɛ	sɛ : canari
I	i	si : père
o	o	bo : forêt
ɔ	ɔ	klɔ : village
u	u	su : oreille
$\tilde{a}$	an	kpan : crier
$\tilde{\varepsilon}$	ɛn	mɛn : avaler
$\tilde{\imath}$	in	sin : passer
$\tilde{\mathfrak{o}}$	ɔn	tɔn : cuire
$\tilde{u}$	un	sun : pleurer

Several problems are as automatic speech processing is a difficult area, and currently not fully resolved:

- There are no separators, silences between words, comparable to whites in written language.
- Each basic sound (also called phoneme) is amended by context (close): the phoneme that precedes it, and which ever succeeds him. This is due to coarticulation: the fact that when a phoneme is pronounced, the pronunciation of the next phoneme is prepared by a movement of the vocal tract.
- The speech has a very high variability: intra-speaker variability, due to the speech style (singing voice, shouted, whispered, hoarse, husky, under stress, stuttering etc.); speaker variability (different stamps, male voices, female, children's voices etc.); variability caused by signal acquisition (type of microphone) or the environment (noise, crosstalk etc.).

These problems will largely be resolved if we opt for the isolated word recognition.

Table 2. "Baoulé" language consonants.

API. /A. P. A. [1]	Orthographe baoulé	Exemples
b	b	bo : frapper
c	c	cɛ : partager
d	d	di : manger
f	f	fa : prendre
g	g	gale : indigo
gb	gb	gbo : cuisine
j	j	ja : marier
k	k	ka : mordre
kp	kp	kpɔ : détester
l	l	lɔ : là-bas
m	m	man : donner
n	n	nin : mère
ŋ	ny	nyan : gagner
p	p	pepe : singe
r	r	tra : attraper
s	s	se : dire
t	t	to : acheter
v	v	nvan : odeur
w	w	wu : mourir
j	y	ya : douleur
z	z	nzan : bangui

3. Methodology

3.1. Fourier Transform

We are working in all square integrable functions $L^2(\mathbb{R})$, which is a pre-Hilbert, and we consider the orthogonal family of sine $\{e_\omega = t \to e^{i\omega t} / \omega \epsilon \mathbb{R}\}$. The Fourier transform F (or $TF(f)$) of f some Function allows a projection on the vector space generated by the sinusoidsfirst by processing the component of the projection on each sinusoid given by the scalar produce, and not forgetting the complex conjugation compared to second place :

$$F(\omega) =< f \mid e_\omega >= \int_{-\infty}^{+\infty} f(t) e^{i\omega t} dt.$$

This transform will never serve us rebuild the signal, in our study, but more qualitatively to examine the contributions of each frequency range in the voice signal by studying the spectrogram, which is actually a three-dimensional representation showing the magnitude in the time-frequency plane. However, we must adapt the continuous representation to the discrete case, which is that of digital computing.

3.2. Discrete Fourier Transform

The signal is represented by samples uniformly sampled in time :$\{f(n) / n \in [\![0, N-1]\!]\}$

[1] Alphabet Phonétique International / Alphabet Phonétique Africain.

What we make of them is a period of signal N to avoid edge effects. The discrete Fourier transform is then given by :

$$F(k)=\sum_{n=0}^{N-1} f(n)e^{\frac{-2i\pi kn}{N}}.$$

Observation frequency is proportional to k/N factor for $k \in [0,\frac{N}{2}]$: If f_e is the sampling frequency, then $f = f_e \frac{k}{N}$. Note that this transform is redundant because the information obtained is concentrated in half the interval $[\![0, N-1]\!]$. Indeed:

$$\forall k \in \mathbb{Z}, F(N-k)$$
$$=\sum_{n=0}^{N-1} f(n)e^{\frac{2i\pi kn}{N}}e^{-2i\pi n}$$
$$=\sum_{n=0}^{N-1} f(n)e^{\frac{2i\pi kn}{N}}=\overline{F(k)}$$

and therefore the module is the same, the phase simply being of opposite sign. If one wants to analyze the signal over a range of 10 kHz, we must take a sample rate of 20 kHz. In practice, it implements the fast Fourier transform (FFT for Fast Fourier Transform) which has a complexity of O ($N\log_2(N)$) instead of O (N^2) by direct calculation, which takes advantage of this redundancy to maximize the calculation.

3.3. Fourier Transform and Convolution

When in continuous representation, the convolution of two functions is defined by:

$$\forall x \in \mathbb{R}, (f*g)(x)=\int_{-\infty}^{+\infty} f(t)g(x-t)dt.$$

By variable change $u = x - t$, the convolution operator is commutative. It is more associative.

The Fourier transform of the convolution of two functions is the usual product of the Fourier transforms.

$$TF(f*g)=F\times G.$$

This result is also valid in discrete representation. It will be used in formant analysis result of the modeling adopted for the voice signal.

3.4. Acoustic-phonetic Decoding Problem

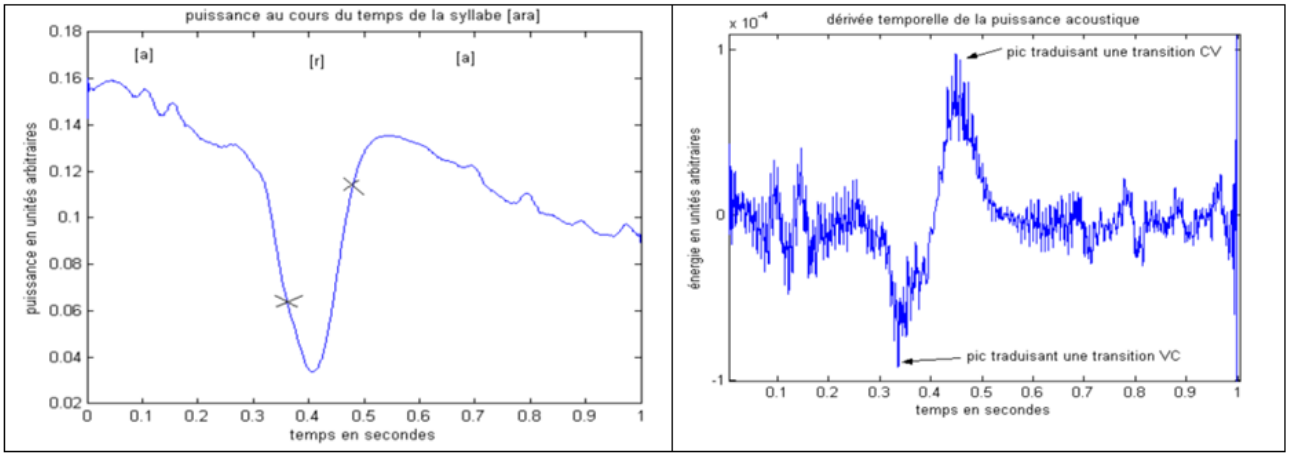
The voice signal is continuous, it is not easy to recognize in the recorded signal the different linguistic elements that are the words, syllables and phonemes. It is this problem called the acoustic-phonetic decoding. Here we have used a method for locating the transitions between vowels and consonants. The syllables for that event are open to qualified. such as the Baule language. It is only a partial solution to the problem, especially as the words to be analyzed must have been previously cleared of initial and final silences.

3.5. Consonant-vowel Transitions (CV) and Vowel-consonant (VC) Detection

To remove as much background noise, we proceed first to a preliminary digital analysis. A filtering has shown that in the conditions of the experiment, the background noise is mainly located below 100 Hz (and mainly 50 Hz frequency at which turn the fans of computers), firstly invokes the **COUPE100** procedure that calculates the signal FFT , cutting harmonics below 100 Hz (taking into account the redundancy) and returns the FFT reverse.

The proposed algorithm (implemented by the **SEPARE** function) is based on the fact that the loudness of the consonants is generally much lower than that of vowels. We first calculate the RMS at each point on a wide enough window to be greater than the possible period of the phoneme (which at this stage is obviously not known). The signal is then derived, having been smoothed to avoid the appearance of peaks due to simple changes, then the derivative itself is smoothed to be easier to treat. Thereafter, we look for $(\varphi - 1)$ most important peaks in absolute value, which therefore correspond to the most brutal varitions (It excludes the edge, which occur aberrant phenomena).

If this peak is negative, the power has fallen sharply, andwe therefore are in the presence of a transitionVC, otherwise there is a CV transition. The graphs below illustrate our method to the syllable [ara].



Empirically estimated (after a series of experiments), that the transition is situated at an instant in the vicinity of the peak where, compared to the preceding phoneme, the signal has lost or gained a power equal to two thirds of the absolute difference in power among the two phonemes. In the graph, the transitions are marked by crosses. The procedure returns separates the moments into which any transitions in the record and a variable "mode" which is 'c'

if the record begins with a consonant, 'v' in the case of a vowel.

3.6. formant Analysis

We chose to recognize by formant analysis vowels and consonants. Before recognizing formants, you must first perform a treatment to reveal more markedly.

Indeed, we find that the magnitude decreases in the spectrogram at a rate of 6 dB per octave. To overcome this attenuation, making it difficult to detect the last formants, it accentuates the voice signal $v(n)$ calculating the magnitude $v'(n) = v(n) - \alpha v(n-1)$ with $v'(0) = v(0)$. The more α, the greater the magnitude is high frequency hand colored. In our experience, we chose α = 2.

In addition, the formants are masked by other effects. Most often, the voice signal $v(t)$ is modeled by the convolution product between the source signal $e(t)$ due to vibration of the vocal cords, and the signal $h(t)$ due to the different resonators.

The homomorphic treatment allows, after calculating a quantity called "cepstrum" to eliminate the influence of the source. To perform the deconvolution, the signal is cut into pieces on which is applied the windowed Fourier transform. then we have $V(\omega) = E(\omega)H(\omega)$ and we take the logarithm of the module (Phase does not matter) to move to an additive representation: $\ln|V(\omega)| = \ln|E(\omega)| + \ln|H(\omega)|$. It then performs inverse transormée, giving the cepstrum: $c(t') = \mathrm{FT}^{-1}(\ln|V(\omega)|) = \mathrm{FT}^{-1}(\ln|E(\omega)|) + \mathrm{FT}^{-1}(\ln|H(\omega)|)$. The cepstrum is expressed in a variable belonging to a sort of time scale distorted by the logarithm called "quefrency". The contribution of the source corresponds to the high quéfrences, and that of the duct to low quéfrences. We must therefore determine the cutoff quefrency from which one has more than the contribution of the source. In practice, simply find quefrency from which the cepstrum understands more pics and remains relatively low. This is called the liftering. Thereafter, it may be possible to recalculate : but to get the spectrogram "corrected", we simply $H(\omega) = \exp(C(\omega'))$.

This processing is implemented by the **ACCENTUE**and **DECONVOL**modules. After correcting the spectrum obtained, each formant is then searched in a frequency band specified as thinly as possible in order to have enough chances to find the formant, which is the highest average magnitude. This operation is performed by the module formants. Estimates of formant values (in Hertz) to a man's voice are given below. extreme values have been indicated for each formant. It should be remembered that these values can vary greatly depending on the individual and within the speech.

Voyelle	F1	F2	F3	F4
[i]	250	2250	2980	3280
[e]	420	2050	2630	3340
[ɛ]	590	1770	2580	3480
[a]	760	1450	2590	3280
[u]	290	750	2300	3080
[o]	360	770	2530	3200
[ɔ]	520	1070	2510	3310
[ɑ]	710	1230	2700	3700
[y]	250	1750	2160	3060
[ø]	350	1350	2250	3170
[œ]	500	1330	2370	3310
[ə]	750	1560	2560	3450

Consonne	F1	F2	F3	F4
[m]	300	1300	2300	2770
[n]	350	1050	3200	3470
[ɲ]	360	1000	2500	3300
[l]	360	1700	2500	3300
[r]	550	1300	2300	2700

Once determined the values of formants, RECONNAISSANCE_VOYELLES and RECONNAI SSANCE_CONSONNES procedures will calculate the distance with each formant values in memory. The formant distance is simply a Euclidean distance; between phonemes A and B, it is applicable:

$$d(A,B) = \sqrt{\sum_{i=1}^{4}\left(F_A{}^i - F_B{}^i\right)^2}.$$

3.7. Overall Operation of the Recognition Algorithm

Recognition is orchestrated by the **RECONNAISSANCE** process. The latter takes as argument recording, without beginning and end of pauses and then asks **SEPARATE** positions of phonemes and what type of phoneme (consonant / vowel) is in first place. Thanks tothese data, it may as appropriate, send the phoneme **RECONNAISSANCE _VOYELLES** or **RECONNAISSANCE_CONSONNES**, which act as described above, then returns the number that we decided to associate to each phoneme. The sequence of these numbers is then returned by recognizing the user.

Of course, before starting **RECONNAISSANCE** procedure, you must be going through a learning phase, carried out by **APPRENTISSAGE_VOYELLES**and **APPRENTISSAGE_CONSONNES** scripts.

Three samples are taken for each phoneme in order to optimize the recognition subsequently. The data is then stored in a table in the working directory (which must be the MATLAB current directory), under the names of **FORMANTS_VOYELLES** and **FORMANTS_SONANTES**.

4. Treatment Outcomes

4.1. Separation

The phoneme separation algorithm works very satisfactorily as long as:

- the start and end of silences were deleted;
- the speaker does not blow on the microphone during recording, bringing the signal at saturation

and results in very large peaks that the algorithm takes for a transition

- we are limited to a maximum of six phonemes (beyond, the algorithm can no longer distinguish variations power between two phonemes and fluctuations).

4.2. Vowel Recognition

The recognition program is not sufficiently developed to discriminate flawless final each of the vowels. In particular, it is still relatively depending on the acoustic features of the memory kept vowels; However, once it is maintained at these features, recognition is done perfectly.

Confusion, however, are far from random and are related to the relative position of the vowels on vowel trapezium :

- Group /a/, /ɑ/ is well recognized because of its particular position in the inferior point of the vowel triangle. Nor do we ask in the case of a speaker speaking a variety of French in which the difference between these two phonemes is not very clear.
- /e/ is sometimes confused with /Ə/ what is consistent because the only thing that differentiates on the vowel triangle is their openness.
- /o/ and /u/ also for the same reasons. If their proximity seems first unclear, it is to think that, in the Latin languages, for example in Occitan, in CATALANET Portuguese (also in French: think of the quarrel between "oïstes" and " ouïstes "enmoyen-french), a /o/ weakened to say precisely /u/.
- /ɔ/ and / ɛ / much more rarely confused respectively, again forproximity vowel reasons.
- The group /Ə/, /ø/,/œ/ is very well recognized. Not clearly differentiating these phonemes between them, I can not speak on their recognition.
- Last but not least, confusion between /i/ and /y/ is very interesting : a French learningSpanish will not (at least, at first) articulate /y/ that he will instinctively replace by /i/.

4.3. Recognition of Consonants Sonorants

Recognition of sonorants unfortunately works much less well. This must be primarily due to the fact that our algorithm seeks horizontal formant when in reality they may vary over time in the case of the consonants, and in addition, depending on the preceding vowels and following consonants, transition effects being then much larger than the vowels, which last much longer. We could follow more accurately the formant trajectories, but then it would implement more complex comparison algorithms and expensive computationally. If some consonants are however fairly well recognized as /n/, and /r/ (as to approximately 50%) were dealing with the confusion following:

- /m/ and /n/ are almost always taken for /n/. This again is consistency, because these are the three nasal consonants in French, among which /n/ may be more intense.
- /l/ is often taken for /r/ : this time, it is difficult to explain.

5. Conclusion

When introducing this research, we said that all French phonemes could not fully cover the phonemes used by African languages, since the acoustic units such as /gb/ and /kp/ have no equivalent in French. Our study allowed us to separate the consonants and vowels of African voice signal to have them identified. It is now possible to create an African corpus to join those actually available. This approach provides a gateway to an accelerated social and economic development of sub-saharan Africa.

The approach that we have proposed can however be improved as part of a future work. For best results, several routes are possible: a better determination of the problem parameters (emphasis coefficient, quefrency cut cepstrum etc.), a more precise analysis of the trajectories and formant transitions and a better management account of prosody in speech. In addition, for a language like Baule, speakers must be honed not only in the use of language, but also to the use of IT resources.

References

[1] IN EMACOP : Environnement Multimédia pour l'Acquisition et la gestion de COrpus Parole Dominique Vaufreydaz, Mohamad Akbar, Jean Caelen, Jean-François Serignat HAL Id: inria-00326144 https://hal.inria.fr/inria-00326144 Submitted on 1 Oct 2008.

[2] IN BENAMMAR Ryadh, TraitementAutomatique De La Parole Arabe, Septembre 2012.

[3] Viet-Bac Le , TRAN D. -D. , CASTELLI E. , BESACIER L., SERIGNAT J. -F. (2004), Spoken and written language resources for Vietnamese, LREC 2004, Lisbon, Portugal, 26-28 May 2004.

[4] (nom de(s) auteur(s))Dictionnaire de la langue Baoulé. (Editeur), (Année d'édition), (Lieu d'édition) (Le nombre de page), (Pages consultées).

[5] LE V.-B., BESACIER L. (2005), First steps in fast acoustic modeling for a new targetlanguage: application to Vietnamese, IEEE International Conference on Acoustics, Speech,and Signal Processing (ICASSP'05), Philadelphia, USA, 19-23 March 2005.

[6] TALN 2005, Dourdan, 6-10 juin 2005 Reconnaissance Automatique de la Parole pour des Langues peu Dotées : Application au Vietnamien et au Khmer L. Besacier (1), V.-B. Le (1), E. Castelli (2), S. Sethserey (3), L. Protin (3) (1) Laboratoire CLIPS-IMAG, UMR CNRS 5524, BP 53, 38041 Grenoble Cedex 9, FRANCE (Laurent.Besacier,Viet-Bac.Le)@imag.fr (2) International Research Center MICA, 1 Dai Co Viet, Hanoi, VIETNAM Eric.Castelli@mica.edu.vn (3) Institut de Technologie du Cambodge, Bd de Pochentong BP 86 – Phnom Penh, CAMBODGE Sam.Sethserey@itc.edu.kh , Ludovic.Protin@online.com.kh.

[7] IN Purdue University: ECE438 - Digital Signal Processing with Applications, October 6, 2010.

[8] IN Haute Ecole d'Ingénierie et de Gestion du Canton de Vaud. Département Technologies Industrielles - Signaux et système. 2008 freddy.mudry@gmail.com.

ANNEXES

MICRO

```
function son = MICRO(t,fs);
        AI = analoginput('winsound');
        chan = addchannel (AI,1);
        duree = t;
        set(AI,'SampleRate',fs)
        vraiefs = get(AI,'SampleRate');
        set(AI,'SamplesPerTrigger',duree*vraiefs)
        set(AI,'TriggerType','Manual')
        blocksize = get(AI,'SamplesPerTrigger');
        Fs = vraiefs;
        start(AI)
        Trigger(AI)
        data = getdata(AI);
end
```

APPRENTISSAGE_CONSONNES :

```
clc;
fs=22050;
loadBANDES_SONANTES;
bandes = BANDES_SONANTES;
loadCONSAPP;
[n,k]=size(CONSAPP);
n=floor(n/3);
FORMANTS_SONANTES = zeros(n,4,3);
disp('***Reconnaissance des formants***')
disp(['Prononcer chaque groupe VCV pendant 1 seconde. laisser lesvoyelles déborder sur le temps de l'enregistrement']);
disp('Ctrl-pause pour interrompre le processus');
fprintf('\n');
for i=1:n
        for c=1:3
                en_cours=1;
                while en_cours
                        input(['Prononcez la syllabe ' ConsApp(3*(i-1)+c,:)]);
                        disp('enregistrement...');
                        syl=microphone(1,fs);
                        disp('OK');
                        fprintf('\n');
                        S=SEPARE(syl,3);
                        if(S(2)-S(1))*1000/fs >= 34
                                en_cours=0;
                        else
                                disp('cet enregistrement s''est mal effectué');
                        end;
                end;
                consonne=syl(S(1):S(2));
                FORMANTS_SONANTES(i,:,c)=FORMANTS(COUPE100(consonne),bandes);
        end;
end;
saveFORMANTS_SONANTESFORMANTS_SONANTES;
```

APPRENTISSAGE_VOYELLES :

```
clc;
fs=22050;
n_essai=3;
loadBANDES_VOYELLES;
bandes= BANDES_VOYELLES;
loadVOYELLES;
[n,k]=size(VOYELLES);
FORMANTS_VOYELLES = zeros(n,4,n_essai);
```

```
disp('***Reconnaissance des formants***')
disp(['Prononcer chaque voyelle pendant une seconde et prononcer la série plusieurs fois de suite']);
disp('Ctrl-pause pour interrompre le processus');
disp('Commencez à parler avant d appuyer sur entrée');
fprintf('\n');
for i=1:n_essai
        for v=1:n
                input(['Prononcez la voyelle ' voyelles(v,:)]);
                disp('enregistrement...');
                voy=COUPE100(microphone(1,fs));
                disp('OK');
                fprintf('\n');
                FORMANTS_VOYELLES (v,:,i) = FORMANTS(voy,bandes);
        end;
end;
saveFORMANTS_VOYELLESFORMANTS_VOYELLES;
```

COUPE100

```
function pur= COUPE100(X);
        fs=22050;
        N=length(X);
        Y=fft(X);
        Y(round(101*N/fs):round(N-101*N/fs))=0;
        pur=X-real(ifft(Y));
end
```

RECONNAISSANCE

```
function suite_phonemes = RECONNAISSANCE(X,fi);
        suite_phonemes=zeros(1,fi);
        enregistrement=COUPE100(X);
        n=length(enregistrement);
        [L,mode]=SEPARE(enregistrement,fi);
        L=[1,L,n];
        for i=1:2:fi
                phoneme=enregistrement(L(i):L(i+1));
                if mode=='v'
                        suite_phonemes(i)=RECONNAISSANCE_VOYELLES(phoneme);
                else
                        suite_phonemes(i)=RECONNAISSANCE_CONSONNES(phoneme);
                end;
        end;
        for i=2:2:fi
                phoneme=enregistrement(L(i):L(i+1));
                if mode=='c'
                        suite_phonemes(i)=RECONNAISSANCE_VOYELLES(phoneme);
                else
                        suite_phonemes(i)=RECONNAISSANCE_CONSONNES(phoneme);
        end;
end;
```

SEPARE

```
function [positions,mode]= SEPARE(X,fi);
        positions=zeros((fi-1),3);
        N=length(X);
        fs=22050;
        fenetre_puissance=500;
        fenetre_lissage=400;
        P=zeros(1,N);
        for i=1:N
                tranche=X((max(1,i-fenetre_puissance/2)):(min(N,i+fenetre_puissance/2)));
                P(i)=sqrt(mean(tranche.^2));
        end;
        P=smooth(P,fenetre_lissage);
        D=smooth(diff(P),fenetre_lissage);
```

```
        separation_non_effectuee=1;
        while separation_non_effectuee
                Dplus=abs(D);
                N_D=length(Dplus);
                bord=200;
                Dplus(1:bord)=0;
                Dplus(N_D-bord:N_D)=0;
                for i=1:(fi-1)
                        [val,pos]=max(Dplus);
                        positions(i,2)=pos;
                        signe_pic=sign(D(pos));
                        indexmin=pos;
                        while (signe_pic*D(indexmin)>0) && (indexmin>1)
                        indexmin=indexmin-1;
                end;
                positions(i,1)=indexmin;
                indexmax=pos;
                while (signe_pic*D(indexmax)>0) && (indexmax<N_D)
                        indexmax=indexmax+1;
                end;
                positions(i,3)=indexmax;
                Dplus(indexmin:indexmax)=0;
        end;
        for i=1:(fi-2)
                [val,pos]=min(positions((i:(fi-1)),2));
                pos=pos+i-1;
                echange=positions(i,:);
                positions(i,:)=positions(pos,:);
                positions(pos,:)=echange;
        end;
        if D(positions(1,2))*D(positions(2,2)) < 0
                separation_non_effectuee = 0;
        else
                pic_gauche=positions(1,2);
                pic_droit=positions(2,2);
                [val,point_critique]=min(Dplus(pic_gauche:pic_droit));
                point_critique = pic_gauche + point_critique;
                D(1:point_critique)=0;
        end;
end;
if D(positions(1,2)) < 0
        mode='v';
else
        mode='c';
end;
positions_exactes=[];
for i=1:(fi-1)
        gauche=positions(i,1);
        milieu=positions(i,2);
        droite=positions(i,3);
        hauteur=abs(P(droite)-P(gauche));
        if D(milieu) < 0
                P_tiers=P(droite)+hauteur/3;
        else
                P_tiers=P(droite)-hauteur/3;
        end;
        [val,pos]=min(abs(P(gauche:droite)-P_tiers));
        positions_exactes(i)=pos+gauche-1;
end;
positions=positions_exactes;
   end
```

RECONNAISSANCE_VOYELLES

function RangVoyelle = **RECONNAISSANCE_VOYELLES** (voyelle);

```
        load FORMANTS_VOYELLES;
        Nvoy=length(FORMANTS_VOYELLES);
        load BANDES_VOYELLES;
        bandes= BANDES_VOYELLES;
        F= FORMANTS(voyelle,bandes);
        D=zeros(3,Nvoy);
        for i=1:Nvoy
                for j=1:3
                        D(j,i)=sqrt(sum((F-formants_voyelles(i,:,j)).^2));
                end;
        end;
        Candidats=zeros(3,2);
        for i=1:3
                [dmin,RangVoyelle]=min(D(i,:));
                Candidats(i,1)=dmin;
                Candidats(i,2)=RangVoyelle;
        end;
        [dminimini,posultime]=min(Candidats(:,1));
        Candidats
        RangVoyelle=Candidats(posultime,2);
   end
```

RECONNAISSANCE_CONSONNES

```
function RangConsonne = RECONNAISSANCE_CONSONNES (consonne);
        fs=22050;
        fmax=fs/2;
        load FORMANTS_SONANTES;
        load BANDES_SONANTES;
        bandes= BANDES_SONANTES;
        Nson=length(formants_sonantes);
        F=FORMANTS(consonne,bandes);
        % Distance euclidienne avec les autres sonantes
        D=zeros(3,Nson); %vecteur contenant les distances
        for i=1:Nson
                for j=1:3
                        D(j,i)=sqrt(sum((F-formants_sonantes(i,:,j)).^2));
                end;
        end;
        %recherche du meilleur candidat
        Candidats=zeros(3,2);
        for i=1:3
                [dmin,RangConsonne]=min(D(i,:));
                Candidats(i,1)=dmin;
                Candidats(i,2)=RangConsonne;
        end;
        [dminimini,posultime]=min(Candidats(:,1));
        RangConsonne=Candidats(posultime,2);
        Candidats+12
        RangConsonne=RangConsonne+12;
end
```

ACCENTUE

```
function X= ACCENTUE(x);
        alpha=2;
        n=length(x);
        X(1)=x(1);
        X(2:n)=x(2:n)-alpha*x(1:n-1);
   end
```

DECONVOL

```
function [Y,taillefft]= DECONVOL (signal,fs,fenetre,n0)
        nbfram=size(signal);
        nbfram=nbfram(1);
        frameparfenetre=round(fs*fenetre/1000);
```

```
        recoupframe=round(frameparfenetre/2);
        hamming=0.54-0.46*cos(2*pi*(0:frameparfenetre-1)'/(frameparfenetre-1));
        fftsize=2.^ceil(log2(frameparfenetre));
        sgrang=2:round(fftsize/2)-1;
        Y=zeros(round(fftsize/2)-2,1);
        for i=1:(frameparfenetre-recoupframe):(nbfram-frameparfenetre);
                sfen=signal(i:min(i+frameparfenetre-1,end));
                sfen=(accentue(sfen'))';
                sfen=sfen.*hamming;
                sfen=ifft(log(abs(fft(sfen,fftsize))),fftsize);
                sfen(1)=[];
                sfen(n0+1:end-n0)=0;
                xf=abs(exp(fft(sfen,fftsize)));
                Y=[Y xf(sgrang)];
        end
        Y=Y(:,2:end);
end
```

FORMANTS

```
function F= FORMANTS(phoneme,bandes);
        phon=phoneme(1:3:end);
        F=zeros(1,4);
        fs=22050/3;
        fmax=fs/2;
        Tfenetre=23;
        quefrence=floor(1000*length(phon)/fs);
        S= DECONVOL(phon,fs,Tfenetre,quefrence);
        magnitude=mean(abs(S'));
        Nf=length(magnitude);
        B=min(round(bandes*Nf/fmax),Nf);
        i=1;
        while i <= 4
                b1=B(i,1);
                b2=B(i,2);
                [val,numpos]=max(magnitude(b1:b2));
                numpos=numpos+b1-1;
                F(i)=round(numpos*fmax/Nf);
                if (i==2) && ((F(2)-F(1)) <= 410)
                        if F(2) <= 763
                                F(1)=F(2);
                        end;
                        B(2,1)=b1+1;
                        i=1;
                end;
                i=i+1;
        end;
end
```

Constants used by the program:

BANDES_SONANTES and BANDES_VOYELLES : 4 rows and 2 columns matrices containing the frequency bands in which to search the formants. These values are adjusted directly by the user by means of MATLAB interface. A readjustment is indeed necessary if we want to recognize a female voice.

VOWELS and CONSAPP: column vectors containing the strings to be displayed during the learning procedures.

FORMANTS_VOYELLES and FORMANTS_SONANTES: three-dimensional array containing three samples for each formant, automatically generated by the program after learning procedures.