SEPARATION OF SPEECH SIGNAL FROM COMPLEX AUDITORY SCENES

Laura Ottaviani and Davide Rocchesso

Dipartimento di Informatica
Università degli Studi di Verona
Strada Le Grazie, 15 - 37134 Verona, Italy
ottaviani@sci.univr.it, rocchesso@sci.univr.it

ABSTRACT

The hearing system, even in front of complex auditory scenes and in unfavourable conditions, is able to separate and recognize auditory events accurately. A great deal of effort has gone into the understanding of how, after having captured the acoustical data, the human auditory system processes them. The aim of this work is the digital implementation of the decomposition of a complex sound in separate parts as it would appear to a listener. This operation is called *signal separation*.

In this work, the separation of speech signal from complex auditory scenes has been studied and an experimentation of the techniques that address this problem has been done.

1. INTRODUCTION

The word *Auditory Scene Analysis (ASA)* was introduced by Bregman to denote the psychoacoustic field founded by him with the objective of understanding how the auditory system and the human brain process the *complex auditory scenes*. In order to know what it means, we can refer to Scheirer [1], who, with reference to McAdams and Yost, defines *auditory image* as what a subject, listening to many sources simultaneously, perceives as a single source. Infact, a sound set is divided in auditory images that the listener can imagine to hear independently. A sound group perceived as made of auditory images is called *auditory scene* and the perceptive process applied to an auditory scene is called *auditory scene analysis*.

Recently, a new research field was introduced, the *Computational Auditory Scene Analysis (CASA)*, aimed at developing computational systems able to simulate what the Auditory Scene Analysis does.

The CASA systems can be classified as data-driven or prediction-driven. The *data-driven* approach is based only on the input signal features. The systems based on this approach are called *bottom-up*, because, analyzing the information collected at low level from the physical features of the sound, they build, at an higher level, conceptually abstract descriptions. On the other hand, the *prediction-driven* approach defines the *top-down* systems. The latter approach addresses the problem of the *auditory illusions*, a perceptive phenomenon that isn't explainable with bottom-up processes. We define an *illusion* as the twisted correspondence from appearance and reality and an *auditory illusion* as the particular illusion that involves the auditory system. Our project could be defined as a bottom-up system.

One of the goals of the computational auditory scene analysis is the *speech segregation*, i.e. the process that separates speech signals from other interferring sounds. One other approach addressing this problem is the *blind source separation*. It is a statistical technique used for separating concurrent sound sources and it isn't inspired by auditory processes as the techniques cited previously. The blind source separation relies on two main assumptions: the signal sources must be statistically independent and the mixing process must be linear. Infact, the signal sources distributions aren't known a priori and not even the mixing process nature. The blind source separation algorithm tries to invert the mixing process in order to reconstruct the single components.

In [2], we can find a comparison between the two techniques mentioned so far, i.e. the computational auditory scene analysis and the blind source separation. The authors submitted the same input sounds, made of speech and noise signals simultaneously, to the two different algorithms, chosen as rappresentative of the two approaches. Analysing the collected results, they concluded that the CASA system provides poor results in the case of large band noises, such as random or vocal noises. They noticed that the system performance is particularly bad if the noise signal is a female speech. This is due to some errors in the pitch-tracking procedure. The article underlines that the CASA approach performances reflect the human capabilities, in the case, for example, of the "cocktail party" effect. Infact, the hearing system capabilities for the signal separation are limited when it is subject to many concurrent auditory signals; for example, in a cocktail party we can listen to different sound sources, but we can pay attention only to one of them. The same limitation can be noticed in the CASA system performance too. On the other hand, the blind source separation is a powerful technique, but only if the following conditions are respected: the number of the mixing signals must be equal to the number of the available sources and the sources must be perfectly aligned in time. These conditions are quite restrictive but, if respected, they can give satisfactory results. On the contrary, the CASA algorithms require less conditions and, therefore, they are more flexible. Since we want to be able to separate signals in whatever condition, not being subject to restrictions, we studied the computational auditory scene analysis approach, in order to exploit these positive results.

2. THE PROPOSED APPROACH

For the prototype implementation, we used particular techniques, each one addressing a specific aim. We divided the structure in two main connected blocks: the first one implements the *pitch analysis*, while the second one achieves the *resynthesis*. The pitch analysis block is based on the Computational Auditory Scene Analysis techniques. Using the correlogram and summary correlogram rep-

resentations, it estimates the pitch of the input signal.

The *correlogram* represents sound as a three dimensional function of time, frequency and periodicity. A cochlear model serves to transform a one dimensional acustic pressure into a two dimensional map of neural firing rate as a function of time and place along the cochlea. A third dimension is added to the representation by measuring the periodicities in the output from the cochlear model [3]. In order to compute the pitch, by means of the correlogram, we compute the *summary correlogram*, summing across frequency channels, in each frame.

In our prototype, the analysis phase extracts, from the sound, the data needed by the resyntesis phase.

The second block is based on the spectral analysis and the inverse Fourier Transform and it resyntesizes the signal, trying to keep only the speech signal and to eliminate the other components.

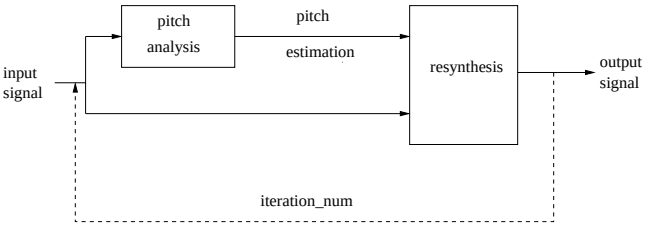

Figure 1: *General diagram of the structure.*

Fig. 1 represents the general diagram of the prototype structure. The input of the pitch analysis block is the signal containing the complex auditory scene to be analysed. After the pitch estimation of the components, the first block transmits the data to the second block that resythesizes, from the original acoustic scene, only the speech signal.

We will examine in detail each block in the following sections.

2.1. Signal analysis

In fig. 2 we report the diagram of the pitch analysis block.

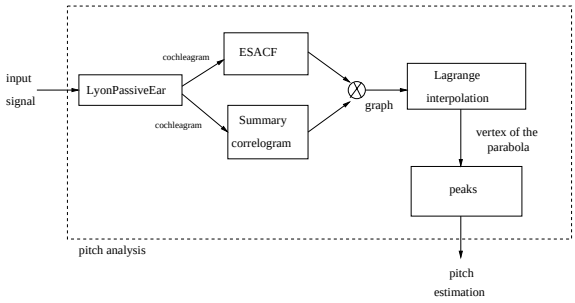

Figure 2: *Diagram of the pitch analysis block.*

The first block of the diagram contains the LyonPassiveEar function of the Auditory Toolbox ([4]) which, based on the Lyon's passive model, computes the firing rate along the auditory nerve due to the input signal. Therefore, it computes the data represented by the *cochleagram*. The cochleagram is, then, passed to the next two blocks of fig. 2 that compute, respectively, the Enhanced Summary Autocorrelation Function (ESACF) and the summary correlogram.

The computation of the ESACF, based on what presented in [5], is composed of two operations: the computation of the Summary Autocorrelation Function (SACF) and, following, the Enhancing. We perform these two subsequent operations for each frame of the input cochleagram. For the computation of the cochleagram, we used an Hamming window of 1024-point and an hop-size of 256-point, i.e. $\frac{1}{4}$ of the window lenght. The reference sample rate is 16 KHz. The SACF is calculated by means of the "generalized autocorrelation" but, as opposed to [5], we examine all the channels of the cochleagram, instead of only two channels, in order to imitate the method applied by human hearing system and to estimate the pitch more accurately. The SACF is obtained by the following equation:

$$\begin{aligned} SACF &= \sum_{i=1}^{N\_channels} IDFT(|DFT(x_i)|^k) \\ &= IDFT(\sum_{i=1}^{N\_channels} |DFT(x_i)|^k) \end{aligned} \quad (1)$$

where $N\_channels$ is the number of frequency channels in the cochleagram and k is the compression factor of the amplitudes in the frequency domain. For $k = 2$ we have the standard autocorrelation. A good choice is $k = 0.67$, since we noticed that this value allows us to obtain a graph which best shows the pitch estimation.

For each frame and each frequency channel of the cochleagram, after the computation of the SACF, the Enhancing operation gives prominence to the peaks which are candidates for the pitch estimation, getting rid of the redundant or false information. In this way, we can obtain the ESACF.

We have already introduced the way to compute the summary correlogram. More precisely, if we define $A(i, j, \tau)$ as the autocorrelation of the frame j, for the frequency channel i with lag τ and we consider N frequency channels, we define the summary correlogram $s(j, \tau)$ for the frame j and lag τ as follows:

$$s(j, \tau) = \sum_{i=1}^{N} A(i, j, \tau) \quad (2)$$

In order to highlight the useful peaks for the pitch estimation, we multiply the summary correlogram by the ESACF.

Since we wanted a thorough estimation of the pitch, avoiding that unnecessary peaks mask the actual signal peaks, we decided to limit the range of the pitch search, according to the typical frequency ranges of the voice, reported in table 1 coming from [7].

Subjects	Typical frequency ranges (Hz)
Adult male	85-155
Adult female	165-255
Child (10 years old)	208-259
Baby (12 months)	247-410

Table 1: *Typical frequency ranges of the voice.*

According to these table and to tests, we considered the range between 70 Hz and 250 Hz. Although from the table it seems that the best range could be between 80 Hz and 260 Hz, we noted by tests that the best result is achieved with the range between 70 Hz and 250 Hz. Defining this range allows us to increase reliability in the pitch estimation.

Although we used the graph of the multiplication of the ESACF and the summary correlogram together, and the definition of the typical range of the voice, the pitch estimation is difficult, especially, when the lag is closest to zero. Infact, the abscissae axis of the graph represents the lag in seconds and the pitch. measured in Hz, is proportionally to the inverse of the lag. Therefore, high pitches correspond to small lags. For solving this problem we used the *Lagrange interpolation*. It receives the simplified autocorrelation as input and returns the axis of the parabola, i.e. the line orthogonal to the abscissae axis where the vertex of the parabola is. By means of the vertex of the parabola, we can estimate the signal pitch frame by frame.

2.2. Signal resynthesis

The signal resynthesis block resynthesizes the speech signal in the complex auditory scene, starting from the data obtained during the pitch estimation phase.

In fig. 3 the general diagram of the resynthesis block is represented.

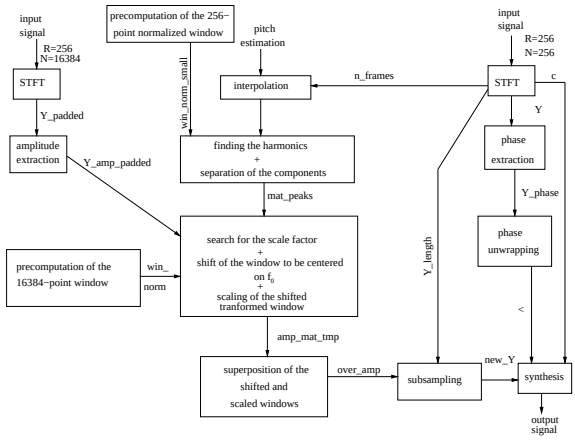

Figure 3: *Diagram of the resynthesis block.*

It gets, as inputs, the signal and the pitch estimation and it returns the resynthesized signal. This block is based on the precomputation of the Fourier transform of the 256-point normalized window and of the padded-16384-point window. The former is used for separating two components with closest frequencies, while the latter is used for reconstructing the amplitude of the signal spectrum.

Observing the fig. 3, we can note that data coming from the pitch analysis phase are not directly used, but are interpolated. Infact, in the pitch analysis phase, we use a 1024-point Hamming window, while, in the resynthesis phase, we use a 256-point Hamming window. Moreover, the hop size, in the second processing phase is not equal to $\frac{1}{4}$ of the window length, but to $\frac{1}{2}$, since we applied a FFT^{-1} technique with both an Hamming and a triangular window. (We will examine the reason for this choice later in the paper.) Therefore, during the resynthesis phase we have more frames than in the pitch analysis one. Therefore, we need to interpolate the data of the pitch estimation, in order to have the same number of frames in both the phases. This interpolation allows us to obtain a pitch function with closest values among two frames.

The values from the interpolation are passed to the block which finds the harmonics for each pitch found in each frame and separates the components with too close frequencies, to avoid the superposition of the main lobes in the frequencies domain. Furthermore, this block gets rid of the redundant harmonics.

Then, we pass the matrix *mat_peaks*, which contains the pitches of each frame and their harmonics, to the block that looks for the scale factor to be applied to the Fourier transform of the normalized padded-16384-point window, in order to reconstruct the amplitude of the signal spectrum. During the search for the scale factor, we wonder also whether an harmonic is meaningful or it is due to random noise. Therefore, if the amplitude (in dB) of the padded STFT of the signal inside the examined bin is lower than a threshold, the related harmonic is considered as meaningless and, so, its value is set to zero. An important topic, during the development of the prototype was to establish the right "meaningfullness" threshold. During the testing phase, we noted that setting the threshold to a value, such as 20 dB, could be a drastic choice, because we don't consider the window envelope in the frequency domain. Infact, if we listen to the output signal, we can hear a "liquid-like" sound. On the contrary, the Fourier transform of the Hamming window slowly decrease outside the main lobe. Therefore, we tried to choose a threshold decreasing proportionally with the distance from the harmonic examined. We made some tests for choosing the slope of the threshold, but we obtained good results only for few input sounds. Since we faced the balance between best quality for a restricted number of files and a lower quality for a wide set of files, we decided to favour the latter one, setting the threshold to a fixed level of 20 dB.

In this resynthesis phase, besides looking for the scale factor, we shifted the Fourier transform of the normalized padded-16384-point window to be centered on the frequency f_0, with f_0 indicating each frequency in the matrix *mat_peaks*. Since the precalculated Hamming window is real and symmetric about zero, also the trasform is real and symmetric and, so, the phase is useless for the computation. Therefore, we extracted the amplitude of the shifted tranformed window and we scaled it by means of the scale factor. The result of this operation is defined *amp_mat_tmp*. We repeated the operations of this block for all the f_0 values of the *mat_peaks* matrix, and each *amp_mat_tmp* was added to the previous ones, as we can see by the block called *superposition of shifted and scaled windows* of fig. 3. In this way, we construct the amplitude of the 16384-point spectrum of the STFT of the resyntesized signal, recorded in the *over_amp* matrix.

The *over_amp* matrix is the input of the *subsampling* block, bacause we need to have a non padded spectrum amplitude, and it returns the *new_Y* matrix. But it contains only the spectrum amplitude of the resynthesized signal. Therefore, it is necessary to compute the spectrum phase. For this reason, from the 256-point STFT of the original signal, we extracted the phase, recorded in *Y_phase*, on which we apply a phase unwrapping technique. This method is recursive, but when the amplitude of a component is 40 dB lower than the maximum, while, in the next frame, it is higher than the threshold, the phase unwrapping starts again. In this way, we consider only the "active" components in each frame.

When we have both the amplitude and the phase of the spectrum, we are able to synthesize the signal. In the prototype, we synthesize the signal by means of additive synthesis based on inverse Fast Fourier Transform, using two types of window: the Hamming window in the frequency domain, since it has a low sidelobe level allowing us to set few frequency points, and the triangular window in the time domain, because, for an exact re-

construction of the signal, the sum of two overlapping windows must be 1 in the overlapping region and, in this region the amplitudes and the phases of the partials must be interpolated by the windows starting from the values of one frame to the ones of the next frame [6].

The *synthesis* block returns the synthesized signal.

2.3. Observations

It is worth mentioning some techniques and original ideas that helped us to implement our prototype.

First of all is the utilization of two separate blocks: one for the *pitch analysis* and one for the *resynthesis*. Since our goal consisted on separating audio sources, we thought to divide the whole process in two steps that, first, extract the data from the auditory scene and, second, exploiting these data, reconstruct only the required scene portion, throwing away the rest. This division in two blocks allows to accelerate the processing time. Infact, by estimating the computing complexity, we found a balance between the analysis and the resynthesis phases and, therefore, they can be allocated on two distinct processors, allowing a concurrent elaboration.

As opposed to what proposed in [5], our Summary Autocorrelation Function examines all the frequency channels derived from the cochleogram, instead of using only two channels. This idea was introduced for trying to imitate the manner of how the hearing system perceives audio signals. Using this new function, the system is more accurate in its pitch estimation, even if the algorithm is computationally more complex.

It is interesting, also, the introduction of the summary correlogram in the pitch analysis block, due to non-excellent results from the ESACF. Multiplying the two functions together, we can estimate the pitch with better performance. Even if during the pitch analysis phase we calculate the autocorrelation, channel by channel, both in the ESACF and the summary correlogram, in the former case it is a generalized autocorrelation with $k = 0.67$, while in the latter case it is a standard autocorrelation, and, therefore, it isn't possible to merge the computation in only one operation.

About the resynthesis choices, there are three important points to underline. First, we used a 256-point window and one padded-16384-point window. Such extreme zero padding proved to be beneficial to obtain accurate estimates of the amplitude of partials. This is critical to avoid amplitude-modulated artifacts in signal reconstruction. Another fundamental aspect of the second step is the application of the technique found in [6], that suggests an FFT^{-1}-based additive synthesis with better performances. Therefore, we applied a Hamming window in the frequency domain and a triangular window in the time domain. Infact, during the prototype development, we had a bad signal reconstruction, even without added noise, due to the fact that, when the sinusoids are moving, in the frequency domain, we obtain a more pronunced lobe. Third, we highlight the unwrapping process, that we start again every time a component goes under 40 dB than the maximum value, considering only the "active" components in each frame.

One topic to consider is the prospect to simplify the blocks in order to reduce the computational costs. Infact, we know that some choices made during the prototype development have affected the computational cost, but the prime goal was accuracy. If we want to improve the prototype, accepting some compromises, we could use a SACF computed only on two frequency bands and a padded-8192-point window. But, if we want to reach a good result, without disadvantages, we could develop the CorrelogramArray and the LyonPassiveEar, on purpose, since, in this prototype version, we applied the ones from the Auditory Toolbox, that doesn't directly address the problem we face.

3. CONCLUSIONS

We highlight that the prototype has been implemented using the MATLAB 5.2 system, whose environment has been extended by means of the Auditory Toolbox [4] and the STFT and the inverse STFT functions, called *stft* and *synthesis*, available from
`ftp://ftp.ci.tuwien.ac.at/pub/export/octave/octave-ci.tar.gz`
adapted for the MATLAB system and properly modified to match our requirements.

The prototype was developed as a workbench for testing different analysis and reconstruction techniques. Several optimizations are possible and should be introduced to obtain a robust and efficient implementation.

4. REFERENCES

[1] Scheirer, E. D., "Music-Listening Systems", PhD thesis, Massachusetts Institute of Technology, June 2000.

[2] van der Kouwe, A. J. W., Wang, D. L., Brown, G. J., "A comparison of auditory and blind separation techniques for speech segregation", Technical report: Osu-cisrc-6/99-tr15, Department of Computer and Information Science, The Ohio State University, Columbus, Ohio 43210-1277, 1999.

[3] Slaney, M., Lyon, R. F., "On the importance of time - a temporal representation of sound", In M. Cooke, S. Beet, and M. Crawford, editors, Visual Representations of Speech Signals, pages 95-116, John Wiley, 1993. Available from `http://www.slaney.org/malcolm/pubs.html`.

[4] Slaney, M., "Auditory toolbox, version 2", Technical Report 1998-010, Interval Research Corproation, 1998. Available from `http://www.slaney.org/malcolm/pubs.html`.

[5] Tolonen, T., Karjalainen, M., "A computationally efficient multi-pitch analysis model", IEEE Transactions on Speech and Audio Processing, 8(6):708-716, November 2000.

[6] Freed, A., Rodet, X., Depalle, Ph., "Synthesis and control of hundreds of sinusoidal partials on a desktop computer without custom hardware", In Proceedings of the ICSPAT '93, 1993.

[7] Baken, R. J. "Clinical measurement of speech and voice". Taylor and Francis Ltd., London, 1987.