Behavior Research Methods
2006, 38 (4), 574-578

Oreja: A MATLAB environment for the design of psychoacoustic stimuli

ELVIRA PÉREZ
University of Liverpool, Liverpool, England

and

RAUL RODRIGUEZ-ESTEBAN
Columbia University, New York, New York

The Oreja software package (available from www.liv.ac.uk/psychology/Downloads/Oreja.htm) was designed to study speech intelligibility. It is a tool that allows manipulation of speech signals to facilitate study of human speech perception. A feature of this package is that it uses a high-level interpreted scripting environment (MATLAB), allowing the user to load, break down into different channels, analyze, and select the parts of the signal(s) of interest (e.g., attenuation of the amplitude of the selected channels, addition of noise, etc.).

Psychologists have sought to understand how human listeners understand language, and specifically how the auditory system efficiently processes language in noisy environments. Listeners possess strategies for handling many types of distortion. For example, the "restoration effect" or "illusion of continuity" occurs when an intermittent sound is accompanied by a masking noise that is introduced in the gaps so as to exactly cover the silent spaces, and the intermittent sound is heard as continuous (see Ciocca & Bregman, 1987). Elucidation of these is important for understanding speech intelligibility in noisy environments, designing robust systems for computational hearing, and improvement of speech technology.

Generally, psychologists do not possess specialist skills in signal processing, frequency analysis, acoustics, or computer programming. Similarly, most engineers do not have in-depth knowledge of statistical analysis or cognitive neuroscience. Interdisciplinary research groups have emerged to cope with this problem in an attempt to integrate specialized knowledge. The aim of interdisciplinary speech approaches is to combine different sources of knowledge, attitudes, and skills in order to better understand sophisticated auditory and cognitive systems.

The development of this software has been partially supported by HOARSE Grant HPRN-CT-2002-00276 and a Fulbright scholarship granted to the first author. We acknowledge and thank Dan Ellis, without whom this work could not have proceeded. We thank John Worley, Dimosthenis Karatzas, Harry Sumnall, Julio Santiago, and Martin Cooke for helpful comments and suggestions on earlier versions of the manuscript. Part of this work was presented at the 2005 European Society of Cognitive Psychology Conference, Leiden, The Netherlands. [Oreja may be used freely for teaching or research, but it may not be used for commercial gain without permission of the authors.] Correspondence concerning this article should be addressed to E. Pérez, Department of Acoustic Design, Kyushu University, 4-9-1 Shiobaru, Minami-ku, Fukuoka 815-8540, Japan (e-mail: perez.elvira@gmail.com).

The development of the Oreja software was inspired by several considerations. The first motivation was the need for an interactive and exploratory tool—specifically, an intuitive interface with which users could dynamically interact and which would demonstrate virtually the phenomena found in speech and hearing. Whereas there are different examples of auditory demonstrations, such as *Demonstrations of Auditory Scene Analysis: The Perceptual Organization of Sound*, by Bregman and Ahad (CD included in Bregman, 1990), and more recently, the auditory demonstrations of Yoshitaka Nakajima (available at www.design.kyushu-u.ac.jp/~ynhome/ENG/index.html), these allow the user only to listen, not to explore and manipulate variables of interest.

It is well known that dynamic interaction with an environment can improve learning in any field, especially when such interaction involves the transformations and manipulations of several different parameters. Direct manipulation of parameters can be of help to the novice by decreasing learning times; to the expert, through its speed of action; and to intermediate users, by enabling operational concepts to be retained. Across all experience levels, allowing the user to initiate actions and predict responses (see, e.g., Shneiderman, 1983) reduces anxiety and bolsters confidence. Oreja encourages the user to explore and replicate the range of relevant variables underlying auditory effects such as the perceptual grouping of tones (see, e.g., van Noorden, 1977) or duplex perception (see, e.g., Rand, 1974). With respect to masking, users can filter speech into different bands, select and apply maskers from a menu of noises, and hear the result of their manipulations. Another important aspect of Oreja is its simplicity: It enhances basic features such as the visual display of signals and the parameters most used in speech intelligibility research (e.g., maskers, filters, or frequency bands). The simplicity of Oreja was motivated by the consideration

that too much sophistication can overwhelm intermediate or novice users. Moreover, the different visual representations of the signals help to reinforce complementary views of the data and to provide a deeper understanding of the auditory phenomena.

Another motivation for the development of Oreja was the work of Kasturi, Loizou, Dorman, and Spahr (2002), which assessed the intelligibility of speech with normal-hearing listeners. Speech intelligibility was assessed as a function of the filtering-out of certain frequency bands, termed *holes*. The speech signals presented had either a single hole in various bands or had two holes in disjointed or adjacent bands in the spectrum. In order to further develop this earlier work, we wanted to investigate the intelligibility of speech signals when some of these frequency bands were replaced by sine-wave speech replicas (SWS), a synthetic analogue of natural speech represented by a small number of time-varying sinusoids (see Remez, Rubin, Pisono, & Carrell, 1981). The construction of an intuitive interface would allow user-friendly filtering of speech into different channels, menu-driven application of distortions, and output.

A third source of inspiration was the work of Cooke and Brown (1999): MATLAB Auditory Demonstrations, or MAD. These demonstrations existed within a computer-assisted learning application, which provided the user with the ability to perform interactive investigations of the many phenomena and processes associated with speech and hearing. However, we preferred to build a psychoacoustic tool more focused on the design of psychoacoustic stimuli, with a wide range of possibilities and menus that would be accessible to a large and heterogeneous group of language and speech researchers.

Oreja's main objective is to provide researchers and students with a useful tool for supporting and motivating the design of psychoacoustic experiments. Although a wide variety of professional software exists that offers more audio processing functions than Oreja does, Oreja's advantage and uniqueness is that it brings together under a simple and clear interface the functions that are most important to psychologists.

Description of Oreja

Oreja has been most recently implemented using MATLAB 6.5 (The Matchworks, Inc.) running under Windows XP, a high-level scripting environment that provides functions for numerical computation, user interface creation, and data visualization.

The design of Oreja has been specially oriented for the study of speech intelligibility, supporting the design of acoustic stimuli to be used in psychoacoustic experiments. It has two main windows or interfaces. The first window guides the selection of the signal, and the second window guides the changes performed on the different parts of the original signal or the creation of background noises to mask it. The first window allows loading, breaking down into different channels, analyzing, and selecting parts of the signal; it also allows for labeling of signals and their constituent parts. Moreover, Oreja can load multiple signals and link them together.

The second window has been designed to allow the user to manipulate the loaded signals (or selected parts), in numerous ways (e.g., attenuating the amplitude of the channels selected, or adding noise), and save them in an audio file format.

USING OREJA.M

Oreja can be downloaded from www.liv.ac.uk/psychology /Downloads/Oreja.htm. At present, it is available only for Windows operating systems. Installation is simple; one simply adds the Oreja folder to the MATLAB current directory path. Or, for Oreja to be permanently available, it can be added as a single folder, approximately 1 MB in size, to MATLAB's collection of toolboxes. Help is available in a users' manual, which also contains a glossary that defines the more technical concepts. Once Oreja is in the current directory of MATLAB, the first window can be loaded by typing at the MATLAB prompt the following command: » oreja.

First Window: *Signal*

The main function of this first window is to allow the user to load a signal and select precise portions of it. The file format of these signals should be .au, .wav, or .snd; other formats are not currently supported. Panel 1 (see Figure 1) represents the frequency composition of the signal. The intensity of the signal is represented in the spectrogram with green tones (light green represents greater intensity or energy). Panel 2 shows the signal filtered into 10 frequency channels (by default, unless another number of frequency channels is specified). Selected channels will appear in green. Panel 4 represents the amplitude of the signal across time (waveform of the sound file loaded). In these three panels, time is represented on the *x*-axis and frequency on the *y*-axis; the frequency corresponds to our impression of the highness of a sound. Cursors and a zoom option are provided to facilitate the accuracy of the selection. As the mouse or cursors (shown as vertical lines) are moved around panel 1, the time and frequency under the current location is shown in the top right displays (see number 6 in Figure 1). Cursors from panels 1 and 2 are connected, because both share the same time scale of the signal. However, the cursors from the bottom panel are linked to the cursors from panels 1 and 2 only when the whole signal is represented in the first loaded stage; it is only in this first stage that all the panels represent the whole signal. The spectrogram display has a linear-in-hertz *y*-axis, whereas the filter center frequencies are arrayed on an equivalent rectangular bandwidth (ERB) rate scale, which is approximately logarithmic. A speech signal can be labeled or phonologically transcribed by using the *Transcription* menu shown in panel 3.

The *info* menu contains a popup menu with a detailed user manual, in html format, that also includes a glossary.

Channels and filters. By default, all the channels are selected after the signal is loaded. Different channels can be selected or deselected by clicking on individual waveforms or on the relevant buttons in panel 2 (see number 5 in Figure 1). The whole original signal can be played back,

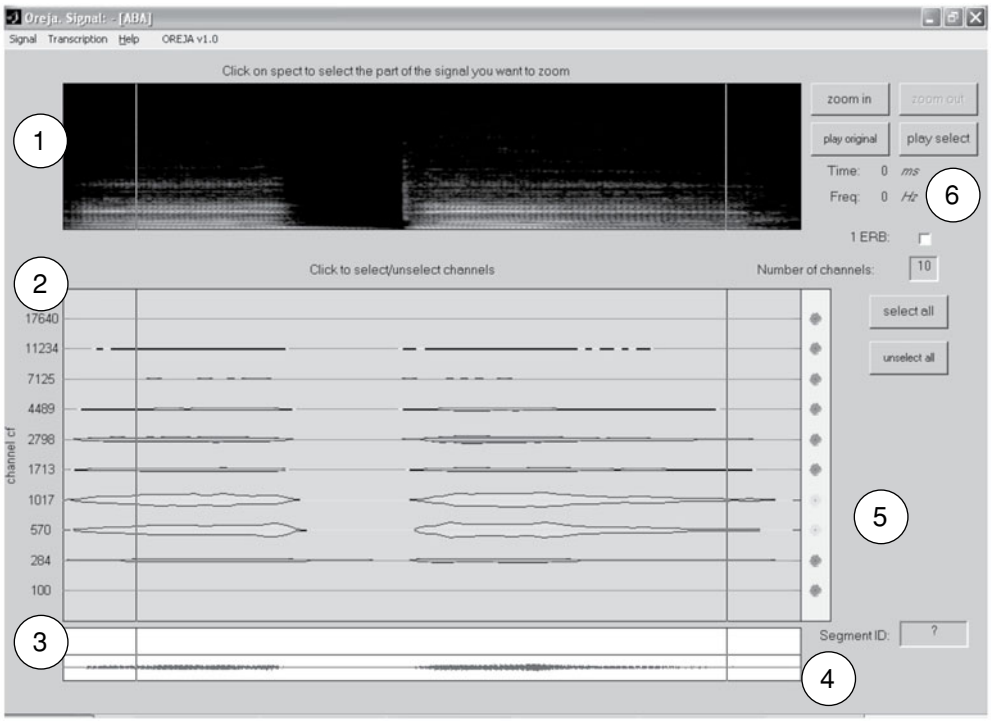

Figure 1. The first window allows the user to select a portion of the signal by time and frequency domain. Three different representations are available: Panel 1 and panel 2 represent the frequency composition of the signal, and panel 4, the amplitude of the signal across time. Panel 3 allows for labeling portions of the signal.

or the channels alone can be selected. For the latter option, an unselected signal does not contribute to the overall output. By selecting or deselecting waveforms, the user can explore various forms of spectral filtering and start designing stimuli. Alternatively, the *select all*/*unselect all* buttons can be used to speed up the selection process. The *play original* button will play back the original signal to provide a comparison to the active signal, if the original already has been distorted. The number of channels among which the signal has been divided can be modified to explore lowpass, highpass, bandpass, and bandstop filtering. The information contained in each band depends on the filterbank applied. A bank of second-order auditory gammatone bandpass filters (see Patterson & Holdsworth, 1996) is used to divide the signal. The center frequencies of the filters are shown on the left side of the display. The distance between their frequency centers is based on the ERB (see Moore, Glasberg, & Peters, 1985), fit to the human cochlea. The default filterbank uniformly covers the whole signal with minimal gaps between the bands. The default bandwidths can be changed; this forces all filters to have a bandwidth of 1 ERB, regardless of their spacing. This option leaves larger gaps between the filtered signal bands for banks of fewer than 10 bands, but the distances between the frequency centers are not changed. Notice that when the signal is filtered by a small number of channels, the default filterbank provides more information than the 1-ERB filters. The suitability of each filterbank depends on the purpose of the experiment.

This first window has been designed to study the ability of the auditory system to process spectral filtering, reduction, and missing data speech recognition. The output of this window can be manipulated in a second window, *Manipulation*.

Second Window: *Manipulation*

After the selection and filtering is done, the *Signal* menu allows access to the *Manipulate selection* option, and a second window, *Manipulation*, appears (Figure 2) with the time and frequency domain selections represented in two panels.

As in the first window, the signal can be selected in either the time or the frequency domains. The exact position of the cursors appears in the time and frequency displays. It is possible to accurately select the time portion the user wishes to manipulate by inserting the specific values in the *from*/*to* option of the *global settings* menu.

The *distortions* menu has been designed to explore the effect on recognition of altering spectrotemporal regions or adding maskers to speech. It has been organized into two subgroups. The first subgroup comprises three different types of maskers: *speech*, *tone*, and *noises.* These will affect all the channels (selected or not), but will only mask the time period selected with the *from*/*to* function from the

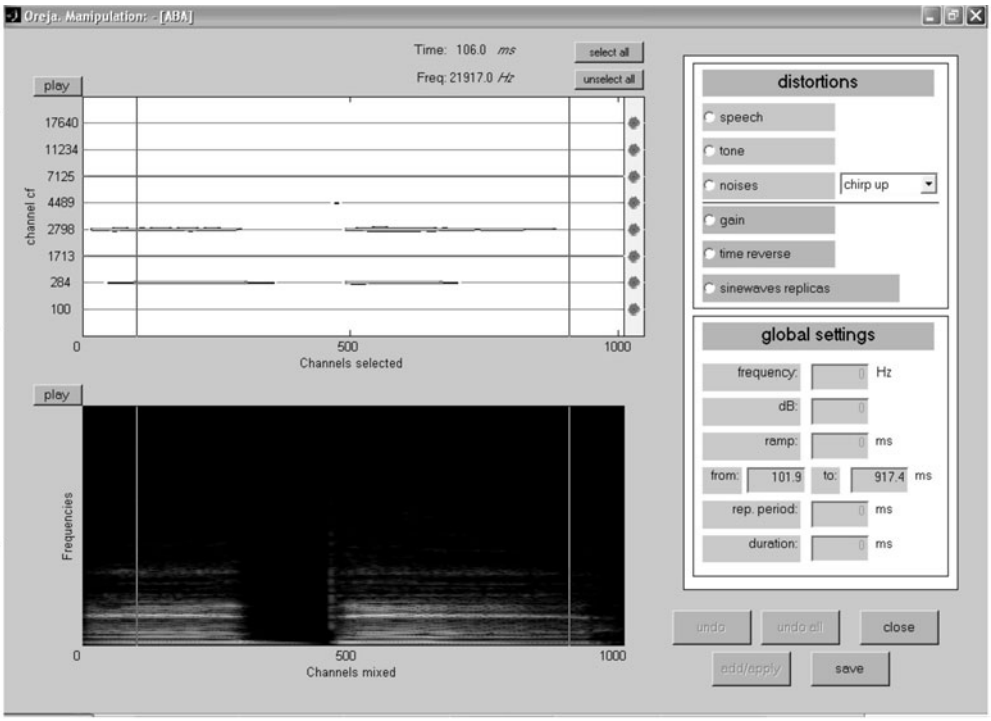

Figure 2. Second window: *Manipulation*. The portion of the signal selected in the previous window can be altered in this window. From the *distortions* menu on the right side, different types of distortions can be selected and applied to the signal. From the *global settings* menu, different parameters from the *distortions* menu can be modified or generated.

global settings menu. The *speech* masker is empty by default and has been designed to load speech previously recorded and saved by the user. The *tone* and *noises* maskers can be generated by inserting the appropriate values in the *global settings* menu, or by selecting a specific type of noise from the popup menu. The second subgroup of distortions comprises *gain*, *time reverse*, and *sinewaves replicas*, and they can change the properties, such as amplitude or time direction, of the channels selected, by positioning time-varying waves at the centers of formant frequencies. None of the three maskers can be frequency-band restricted.

In the bottom panel (see Figure 2) the spectrum of the selected portions can be visualized in combination with the manipulations added. Again, the user has the choice to play back the original signal or the portion of the signal selected with the added distortions.

Distortions. The *distortions* menu displays the stimuli that can be added, subtracted, or applied. Notice that only the global settings that are relevant to each specific stimulus are enabled. As described above, the signal loaded can be masked with speech, a tone, or noises.

Speech. This option mixes the signal selected with streams of speech or with other signals. The specific inputs depend upon the kinds of stimuli the user wants to design. The advantage of this option is that the user can select and manipulate a speech signal and mix it later with another signal to create, for example, a "cocktail party effect"—an effect that addresses the matter of attending to a single voice and ignoring background noise (see Cherry, 1959)—or the user can save the signal and use it at a later date.

Tone. A sinewave tone is generated. The user can select a specific frequency, duration, starting and ending point, ramp, and the repetition rate, if the tone stops and starts at a regular rate.

Noises. This menu contains five stimuli: *white noise*, *brown noise*, *pink noise*, *chirp up*, and *chirp down*. Some of the parameters of all these stimuli can be changed within the code that generates them (.m files from the *noises* folder), or within the *global settings* menu. Notice that stimuli, like bursts, can be generated by selecting a type of noise (e.g., *pink noise*), and setting its duration.

Transformations. A choice of the following three transformations may be applied to the selected channels.

Gain. Adjusts the amplitude level in decibels, applied to the channels selected.

Time reverse. Inverts in time the selected channel.

Sinewaves replicas. Sinewave speech is a synthetic analogue of natural speech, represented by a small number of time-varying sinusoids.

Global settings. General settings that can modify some of the parameters of the signal, channels, or maskers are (1) *frequency*, (2) *amplitude*, (3) *duration*, (4) *from*/*to*, (5) *ramp*, and (6) *repetition period*.

Finally, Oreja allows the user to save the manipulations that have been done, undo the last manipulation, undo all, or annotate the signal.

CONCLUSION

The Oreja software provides a fertile ground for interactive demonstrations and a quick and easy way of designing psychoacoustic experiments and stimuli. Dynamic interaction with acoustic stimuli has been shown to aid learning and provides a better understanding of auditory phenomena (see Cooke, Parker, Brown, & Wrigley, 1999). Due to its user-friendly interface and small processor requirements, Oreja is useful for a broad range of research applications.

REFERENCES

Bregman, A. S. (1990). *Auditory scene analysis*. Cambridge, MA: MIT Press.

Cherry, C. (1959). *On human communication*. Cambridge, MA: MIT Press.

Ciocca, V., & Bregman, A. S. (1987). Perceived continuity of gliding and steady-state tones through interrupting noise. *Perception & Psychophysics*, **42**, 476-484.

Cooke, M. P., & Brown, G. J. (1999). Interactive explorations in speech and hearing. *Journal of the Acoustical Society of Japan*, **20**, 89-97.

Cooke, M. P., Parker, H. E. D., Brown, G. J., & Wrigley, S. N. (1999). *The interactive auditory demonstrations project*. Presented at ESCA, 1999 Eurospeech Proceedings, Budapest, Hungary.

Kasturi, K., Loizou, P. C., Dorman, M., & Spahr, T. (2002). The intelligibility of speech with "holes" in the spectrum. *Journal of the Acoustical Society of America*, **112**, 1102-1111.

Moore, B. C. J., Glasberg, B. R., & Peters, R. W. (1985). Relative dominance of individual partials in determining the pitch of complex tones. *Journal of the Acoustical Society of America*, **77**, 1853-1860.

Patterson, R. D., & Holdsworth, J. (1996). A functional model of neural activity patterns and auditory images. In W. A. Ainsworth (Ed.), *Advances in speech, hearing & language processing* (Vol. 3, pp. 551-567). London: JAI.

Rand, T. C. (1974). Dichotic release from masking for speech. *Journal of the Acoustical Society of America*, **55**, 678-680.

Remez, R. E., Rubin, P. E., Pisono, D. B., & Carrell, T. D. (1981). Speech perception without traditional speech cues. *Science*, **212**, 947-950.

Shneiderman, B. (1983). Direct manipulation: A step beyond programming languages. *IEEE Computer*, **16**, 57-69.

van Noorden, L. P. A. S. (1977). Minimum differences of level and frequency for perceptual fission of tone sequences ABAB. *Journal of the Acoustical Society of America*, **61**, 1041-1045.

(Manuscript received June 2, 2005;
revision accepted for publication July 23, 2005.)