

# SPDEMO – A NOVEL SOFTWARE TOOL FOR TEACHING MULTIMEDIA SIGNAL PROCESSING

Yair Moshe, Nimrod Peleg, Nadav Cohen

Signal and Image Processing Laboratory (SIPL)

Department of Electrical Engineering, Technion – Israel Institute of Technology

Technion City, Haifa 32000, Israel

<http://www.sipl.technion.ac.il>

**Abstract** The move from analog to digital media in recent years has given rise to the flourishing of digital signal processing techniques. In addition to the many benefits of this flourishing, the plenitude and sophistication of these techniques has caused learning and teaching the essence of this subject to be a more difficult task.

In this paper a software tool for teaching multimedia signal processing is presented. This software tool is intended for assisting teaching and demonstrating speech and audio signal processing techniques. Using this tool one can select from a variety of techniques and parameters in a straightforward and simple way. The results of this processing can be played and viewed, numerically and graphically, in order to ease the analysis of the data. The purpose of this tool is to enable easy experimentation and demonstration of different algorithms and thus improve the quality of teaching.

**Index Terms**— education software; education tool; computer based education; computer aided instruction; digital signal processing; speech processing; audio processing; speech tool; audio tool; audio digital watermarking

## I. INTRODUCTION

Digital signal processing (DSP) techniques have become an important part of many modern systems. The rapid evolution of these techniques in recent years has created many sophisticated and detailed models and algorithms. This sophistication and plenitude of details could be difficult to grasp for a typical student with a limited amount of time to invest. Teaching concepts in this subject is indeed a challenging task.

As noted in [1], the fundamental concepts of digital signal processing are highly mathematical and abstract, yet extremely practical and useful. Observing the practical aspects of DSP is beneficial for students and improves the learning experience. Effectively demonstrating signal processing techniques can surely contribute to teaching this subject. Moreover, according to the authors' experience, "learning by doing" has a considerable added value to "learning by observing". One of the great advantages of CBE (Computer Based Education) is the possibility of interaction. The possibility of giving input and getting feedback can sometimes truly bring theory into practice and makes people learn [2].

One way of seeing "DSP in action" is writing an implementation of a DSP algorithm in a high level programming language like C. This solution is usually not practical as an aid for teaching this subject because writing the code takes too much time, the programmer is exposed to many unnecessary details and usually there are no easy-to-use visualization means.

The key for understanding any non-trivial concept is abstraction. Abstraction is the process of picking out only the important details out of the wealth of details. That's the process every one of us makes while approaching a complex theory, technique or algorithm. What differentiates one abstraction from another is choosing which details are more important (and should be learnt first in order to "get the whole picture") and which details are less important (and should be left for a more detailed future learning). Few digital signal processing abstractions have been suggested and are practically used.

One such abstraction is mathematical software packages that allow users to quickly perform complex operation without resorting to writing and debugging lower level programs [1]. Matlab® is such a software package in widespread use that has become almost standard in both academia and industry. Matlab® hides many unnecessary implementation details and enables the user to concentrate on algorithmic aspects of a problem, but this abstraction is sometimes not enough for educational purposes. Implementing an ADPCM audio encoder and decoder, for example, using Matlab® is not an easy task. So a higher level of abstraction is desired.

A higher level of abstraction appeared in recent years in graphical programming languages such as Simulink® or Hypersignal®. Such languages are block-oriented. The user can arrange functional blocks from libraries and connect them to form a block diagram. These languages model and simulate dynamic systems. Amir in [3] and others suggest such systems as a teaching tool. This approach shows an easy to understand decomposition of a system but lacks a similar insight into the algorithm, its parameters and their influence on the overall system output.

The authors feel that a better educational abstraction of a DSP algorithm should concentrate on the parameters and their effect on the system performance. Final as well as

intermediate results should be easy to analyze and compare. A tool that enables these requirements will move the experience of learning to a higher level.

Such tools are already available for very specific research topics. For example, two such tools were presented in [4] for speech recognition and in [5] for LPC and LPC-derived cepstrum.

Digital processing of speech and audio is an excellent topic for CBE. Real-world signals could be easily heard and seen. With this in mind, an educational tool called SPDEMO that is described in this paper was developed (and is still under development) in the Signal and Image Processing Laboratory (SIPL), Dept. of Electrical Engineering, Technion, IIT.

## II. SPDEMO – SPEECH AND AUDIO PROCESSING LEARNING TOOL

The inspiration for SPDEMO, the tool that will be described shortly, is VCDemo (Image and Video Compression Demo) [6]. This software, constructed at the Information and Communication Theory Group (ICT) at Delft University of Technology (TUD), is designed as an image and video compression learning tool. With this software in mind, SPDEMO has been designed as a speech and audio processing learning tool. Note that SPDEMO's subjects of coverage are not limited to compression but to all areas of audio and speech processing. Also note that despite the fact that the inspiration for SPDEMO is VCDemo, SPDEMO has been designed and written "from scratch" with many changes and improvements over its source of inspiration.

The main aim of SPDEMO is to illustrate the basic principles of speech and audio processing and their applications. This tool could be used by a teacher during his/her lecture and can also be used by students to experiment with different parameters or for research purposes. SPDEMO incorporates both classical techniques and state of the art techniques which are in SIPL's research fields of interest.

Using SPDEMO one can select from a variety of techniques and parameters in a straightforward and simple way. The results of applying these techniques can be listened to and are shown both numerically and graphically, in order to ease the analysis of data. SPDEMO's GUI (Graphical User Interface) has been designed in order to encourage the user to experiment with it and this is why both ease of use and simplicity have been a major concern.

Fig. 1 depicts SPDEMO's main screen. One can open any WAV file from disk and examine it in various ways – zoom in to view individual samples, zoom out, view a spectrogram or several spectrograms of the signal, each with different parameters, listen to the signal or listen to an amplified version of it (useful for low-amplitude signals, such as the error between a test signal and a reference signal) and see some useful information about the signal such as sampling rate, number of channels and number of bits per channel.

A user can select from a variety of speech and audio

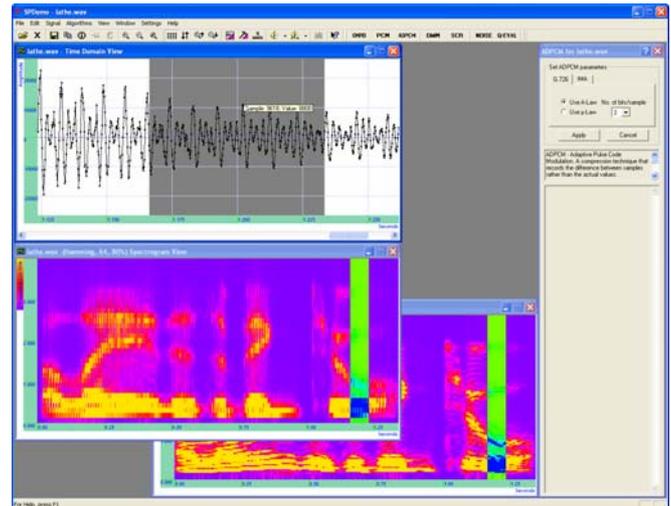


Figure 1: SPDEMO main screen

processing techniques. Each technique opens a separate uniformly designed module window on the right side of the screen. This window allows the application of the technique with different relevant parameters. Changing a module's parameters helps in gaining insight of how well the incorporated algorithm performs under certain parameters. Extreme and even unpractical values are allowed for several parameters. This helps in quickly getting a feeling of "what's right" and "what's wrong".

An application of a module employs the technique with selected parameters and opens an output signal window or windows as well as windows containing intermediate results. In this manner insight is gained of the way the technique works. In addition, results are shown textually and this textual information is also attached to the output signal's information.

A user can keep applying various techniques with different parameters. The signals are disconnected from the modules and this allows easy application of several techniques on the same signal and comparisons between results of different techniques or same technique with different parameters.

Modules which are currently incorporated into SPDEMO or expected soon are:

- A-law and  $\mu$ -Law companding
- Resampling using polyphase filters
- PCM
- ADPCM
- MPEG Audio Layer 3 Compression
- Speech features extraction (e.g., autocorrelation, cepstrum, short time energy, zero crossings rate) and their incorporation into a simple VAD (Voice Activity Detector) and voiced/unvoiced classifier
- Speech enhancement using a Mixture-Maximum model
- Speech packet loss concealment
- Speech time-scale modification using WSOLA (Waveform Similarity Overlap Add)
- Speech time and frequency scrambling

- Digital Watermarking of an audio signal

Two additional utility modules are used in companion with other modules. First is the “Noise and Interference Addition”. This module facilitates an addition of noise or interference to a signal in order to simulate a noisy environment. Gaussian, uniform or periodical (triangle or Sinusoid) signals as well as any WAV file could be added to the signal at a desired SNR. Using this module, various techniques could be analyzed under simulated noisy environments.

The second utility module is “Quality Evaluation”. This module comprises few objective audio and speech quality measures. These measures enable evaluation of the quality of a test signal compared to a reference signal (maybe following the application of one of SPDemo’s incorporated techniques). Currently supported measures are SNR, Segmental SNR, LSD, PESQ (ITU-T recommendation P.862 for Perceptual Evaluation of Speech Quality) and PEAQ (ITU-R recommendation BS.1387 for Perceptual Evaluation of Audio Quality).

All aforementioned features allow easy visualization of algorithm results and the way it works. Audible differences could be comfortably heard and the comparison is easy. All this makes demonstration and analysis using a trial an error approach, very simple activities.

The best way to understand the benefits of using SPDemo is to look at an example. In the following section one of SPDemo’s modules will be briefly presented – the digital watermarking module. This module facilitates digital watermarking of an audio signal and verification of the presence of such a watermark in an audio signal.

### III. DIGITAL AUDIO WATERMARKING AS A CASE STUDY

Recent rapid development in the field of digital media has raised the issue of copyright protection. Digital watermarking offers a solution to copyright violation problems, and also an excellent opportunity to demonstrate a complex process in a very visual and intuitive way. For a reader who is not familiar with this technique, we suggest a short introduction: The watermark is a signature, embedded within the data of the original signal, which in addition to being inaudible to the human ear should also be statistically undetectable, and resistant to any malicious attempts to remove it. In addition, the watermark should be able to resolve multiple ownership claims (known as the deadlock problem), which is achieved by using the original signal (i.e., the unsigned signal) in the signature detection process. In order to meet the above demands, perceptual masking is used, both in frequency domain (using a psychoacoustic model) and in time domain. The added signature is signal dependent, and thus is inaudible as well as adequately robust to attempts to destroy it.

A typical schematic description of such a watermark embedding/verification algorithm is shown in Fig. 2. During the embedding process, the audio signal is divided into segments. For each segment a local key is calculated and

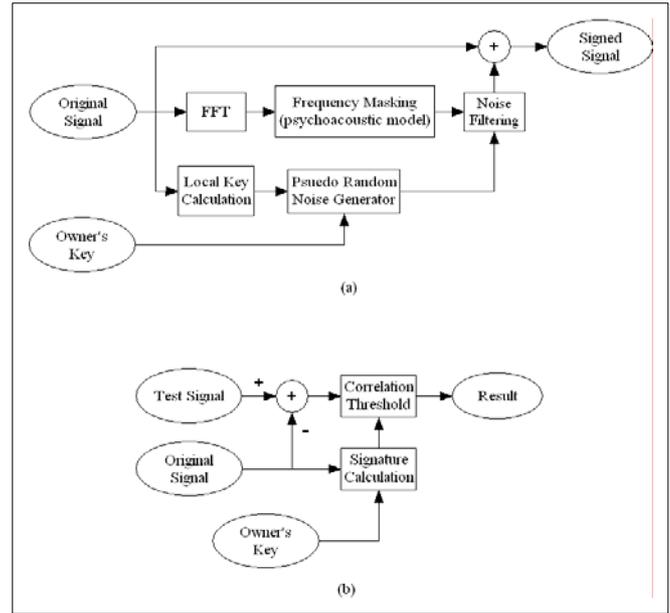


Figure 2: Digital watermarking (a) embedding mechanism (b) verification mechanism

summed up with a general key (independent of the segment) to initiate a pseudo-random noise sequence for the segment. The noise is colored by a filter, which is calculated according to the psychoacoustic model. The colored noise is the watermark which is added to the original signal. More information about this watermarking algorithm (embedding and verification) can be found in [7].

Such a scheme is quite difficult to explain and demonstrate in a classroom without a specific teaching tool. Writing a simulation in Matlab or C is complicated, and changing the parameters of such an application while demonstrating or experimenting with it in real-time is a mission that will require too much of the teacher’s or student’s time.

SPDemo’s digital watermarking module allows the demonstrator and the student to have a close look at the embedding process: the spectrum of the signal, the tonal and non-tonal spectral component, the hearing threshold in quiet, the actual hearing threshold according to the psychoacoustic masking, and the final watermark – all in an interactive way, while pointing at the examined sound segment with the mouse. Fig. 3 is a snapshot from such an interactive session. Fig. 4 shows a more detailed view of one of the available windows. It is also possible to listen to the original signal, the watermarked signal and the watermark itself, isolated from the original audio signal. Many parameters can be changed and examined in an easy way (e.g. the public key, watermark gain, decision thresholds and more), and this makes the learning process clear and even enjoyable.

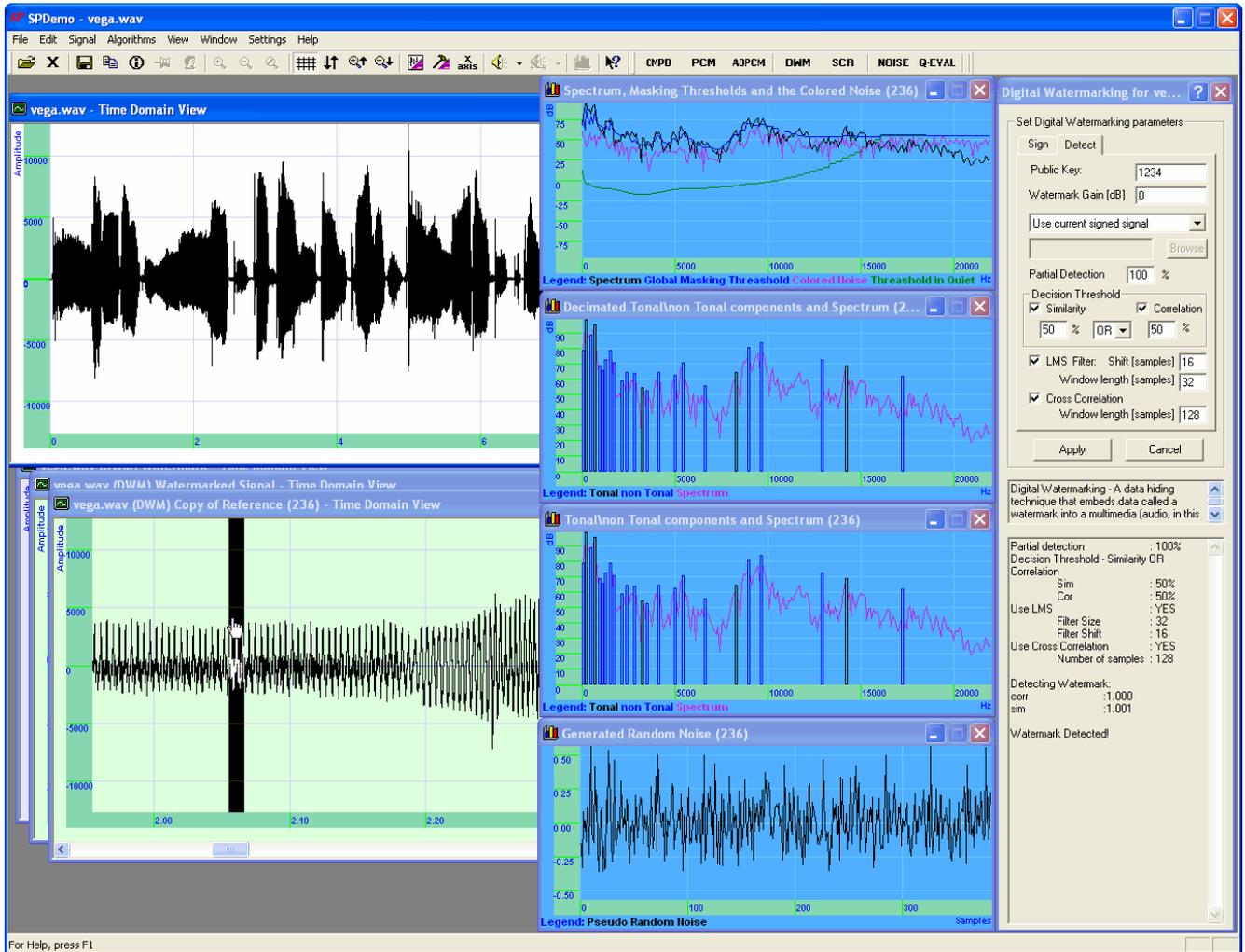


Figure 3: A view of an interactive session using SPDemo's digital watermarking module

#### IV. SPDEMO – DEVELOPER'S SIDE

SPDemo's development process consists of two major parts – creation of the common framework and GUI, and implementation of the different algorithms and integrating them as SPDemo modules.

One of SPDemo's main development concepts is separation between signals and modules. Even from the user's point of view, there is no real linkage between a module and its output signal - a signal can be the source signal of one module, this module's output can be the source of another module, and so on. This concept, of creating an independent entity for each signal, offers great flexibility for both the user and in the development process and simplifies the creation of new modules.

The framework and GUI have been written by an experienced C++ and MFC (Microsoft Foundation Classes) programmer (a student himself), while most of the different algorithms have been implemented and integrated into the

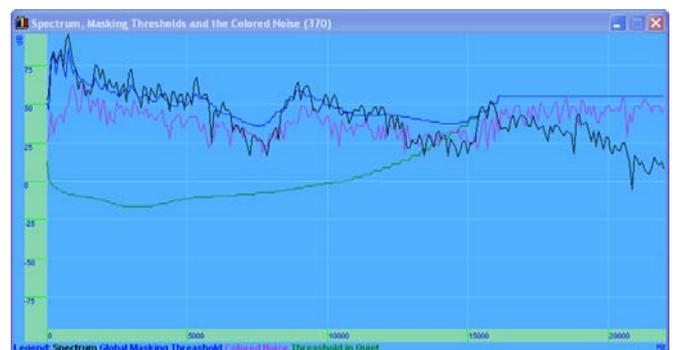


Figure 4: One of the digital watermarking module windows. This window shows intermediate results - spectrum of the signal, threshold in quiet, global masking threshold, and colored noise

application by SIPL's senior students.

Most of the incorporated digital DSP techniques are the product of senior students' projects in the lab. The integration of a DSP technique into SPDemo has been defined as the last phase of some student projects in the lab. This approach, of letting the students themselves, and not the chief programmer, add the techniques implemented by them into SPDemo has some major benefits. The students see the results of their work inside a stable and easy to use working "live" product. This adds to students' motivation and to their dedication to the project.

One problem arising from the demand that students will self-integrate DSP techniques into SPDemo is that many students in the electrical engineering dept., although having a strong electrical engineering foundation, lack knowledge of advanced C++ and MFC programming and have only basic C language programming skills. For that reason, an emphasis has been put on simplifying the integration process as much as possible, thus allowing the basic creation of a new module in a few simple steps while still enabling the use of the framework's many features. This was achieved by designing SPDemo in a flexible way and by the creation of a comprehensive developer's guide.

In addition, the SPDemo framework offers a developer who wishes to add a new module to the application, a set of useful common features. Each module gets an input vector of samples with additional information about the signal properties and returns one or more such vectors with optionally added information. The developer can add to the output signal some extra information which can be displayed in varied visual forms, at the input signal's window or at some separate windows. The module can also use the quality evaluation measures implemented in SPDemo in order to provide the user with an evaluation of the module's performance, by comparing the module's source signal and output signal. The module doesn't deal directly with I/O, graphics or quality evaluation measures. This "bag of features" supplied by SPDemo framework provides flexibility and eases the process of new modules development.

## V. CONCLUSIONS

One of the factors determining the quality of one's learning is the tools he uses. Viewing, listening and trying can undoubtedly improve the effectiveness of learning. The result is a higher level of understanding of the studied topic.

In this paper a software tool for teaching DSP techniques has been presented. The developed tool is based on the concept that students and/or teachers should use an interactive and easy-to-use visual tool as an integral part of the learning process. SPDemo is an implementation of such a tool for the field of digital speech and audio processing techniques.

SPDemo is a valuable tool in training students. It can function as an educational workbench, enabling students to learn from their own experiments or as a tool for a teacher's

demonstrations. SPDemo's analysis features can also be used for research purposes e.g., examination of audio quality.

The latest version of SPDemo can be downloaded from SIPL's website under [8].

## ACKNOWLEDGMENTS

The authors would like to thank the head of the lab, Prof. David Malah, for his support and valuable comments.

The authors would also like to thank the many students who took part in the creation of SPDemo and have helped making it a useful tool (their names can be found in SPDemo's internal documentation).

## REFERENCES

- [1] Zhu J. J., and Johnson W., "Sound demonstration software for teaching DSP," Proceedings of the Twenty-Seventh Southeastern Symposium on System Theory. IEEE Comput. Soc. Press. 1995, pp.450-4. Los Alamitos, CA, USA.
- [2] Rahkila M., and Karjalainen M., "Considerations of computer based education in acoustics and signal processing," FIE '98. 28th Annual Frontiers in Education Conference. Moving from "Teacher-Centered" to "Learner-Centered" Education. Conference Proceedings. IEEE 1998, vol.2. pp.679-84.
- [3] Amir N., "The role of graphical programming languages in teaching DSP," 2000 IEEE International Conference on Acoustics Speech, and Signal Processing. IEEE 2000, vol.6. pp.3514-17
- [4] Nouza J., "System for visual and experimental introduction to basic speech recognition algorithms," Proc of ACL2000 conference. 2000, pp.11-12. Hong-Kong.
- [5] Hong K. S., Salleh S.-H., and Sha'ameri A. Z., "A new developed speech recognition education software in teaching LPC and LPC-derived cepstrum," ICSP'02. 2002 6th International Conference on Signal Processing Proceedings. IEEE 2002, vol.2. pp.1701-4.
- [6] VCDemo – Image and Video Compression tool, <http://www-it.et.tudelft.nl/~inald/vcdemo/>
- [7] Cassuto Y., Lustig M., and Mizrahi S., "Real-Time digital watermarking System for audio signals using perceptual masking," prize winning project in the TI (Texas Instruments) DSP and Analog Challenge, Dallas 2000. <http://www-sipl.technion.ac.il/water/index.shtml>
- [8] SPDemo – Speech and audio processing tool, [http://www.sipl.technion.ac.il/Info/Downloads\\_SPDemo\\_e.shtml](http://www.sipl.technion.ac.il/Info/Downloads_SPDemo_e.shtml)