Proceedings of the Federated Conference on Computer Science and Information Systems pp. 497–502
DOI: 10.15439/2016F296
ACSIS, Vol. 8. ISSN 2300-5963

A Real-Time Audio Compression Technique Based on Fast Wavelet Filtering and Encoding

Nella Romano*, Antony Scivoletto*, Dawid Połap†
*Department of Electrical and Informatics Engineering, University of Catania, Viale A. Doria 6, 95125 Catania, Italy
Email: nromano919@gmail.com, antonyscivoletto@gmail.com
†Institute of Mathematics, Silesian University of Technology, Kaszubska 23, 44-100 Gliwice, Poland
Email: dawid.polap@gmail.com

***Abstract*—With the development of telecommunication technology over the last decades, the request for digital information compression has increased dramatically. In many applications, such as high quality audio transmission and storage, the target is to achieve audio and speech signal codings at the lowest possible data rates, in order to offer cheaper costs in terms of transmission and storage. Recently, compression techniques using wavelet transform have received great attention because of their promising compression ratio, signal to noise ratio, and flexibility in representing speech signals. In this paper we examine a new technique for analysing and compressing speech signals using biorthogonal wavelet filters. In particular, we compare this innovative compression method with a typical VoIP encoding of human voice, underlining how using wavelet filters may be convenient, mainly in terms of compression rate, without introducing a significant impairment in signal quality for listeners.**

***Index Terms*—Wavelet Analysis; Audio Compression; Digital Filters; VoIP; SIP, Quality of Services.**

I. Introduction

SPEECH is a very basic way for people to convey information to each other by means of human voice, within a bandwidth of around 4 KHz. The growth of the computer industry has invariably led to the demand for quality audio data. Analogue audio signals, such as voice speeches, or music, are often represented digitally by repeatedly sampling the waveform and representing it by the resulting quantized samples. This technique is known as *Pulse Code Modulation (PCM)*. PCM is typically used without compression in high-bandwidth audio devices (e.g., in CD players), but compression is essential where the digital audio signal has to be transmitted by means of a communication medium, such as a computer or telephone network [1]. In order to send real-time audio data over a communication link, data compression has be used due to the mismatch with the available link bandwidth [2].

Compression of signals is based on redundancy removal between neighbouring samples or between the adjacent cycles [3]. In data compression, it is desired to represent data by as small as possible number of coefficients within an acceptable loss of quality. Therefore, compression methods rely on the fact that information, by its very nature, is not random but exhibits an intrinsic order and pattern, so that the essence of the information can often be represented and transmitted using less data than would be required for the original signal [2].

Compression techniques can be classified into one of two main categories: *lossless* and *lossy*. Lossless compression works by removing the redundant information present in an audio signal, but preserving its quality and the complete integrity of the data. However, it offers small compression ratios, hence it can be used if we have no stringent requirements; furthermore, it does not guarantee a constant output data rate, since the compression ratio is highly dependent on the input data. On the other hand, one advantage of lossless compression is that it can be applied to any data stream. In lossy coding, the compressed data does not preserve bit-wise equivalence with the original data. The goal of this kind of compression is to maximize the compression ratio or the bit rate reduction, with reduced cost in terms of loss in quality [2].

Compression methods can be classified into three functional categories: *direct methods*, when the samples of the signal are directly handled to provide compression; *parameter extraction methods*, if a preprocessor is employed to extract some features that are later used to reconstruct the signal; *transformation methods*, such as Fourier transform, wavelet transform, and discrete cosine transform. In this latter, the wavelet transform is computed separately for different segments of the time-domain signal at different frequencies. This makes wavelet filtering good for signals having high frequency components for short duration and low frequency components for long duration, such as images, video frames and speech signals [3].

In this paper we show an innovative technique to process and compress an audio signal using a 3.7 biorthogonal wavelet filter, by using thresholding techniques in order to eliminate some insignificant details of the signal, then obtaining a lossy compression that allows us to significantly reduce audio bit-stream length, without compromising the sound quality. While such a technique has been selected because of its remarkable performances, due to the intrinsic nature of the implemented wavelet transforms and filters, it is also possible to implement the coding and decoding pipeline directly on hardware. Therefore, the proposed system not only constitutes an outperforming compression software, but also a possible hardware interface. The latter can be thought both as a consumer-side phone-box or as a provider-side switchboard integrated system.

II. VoIP Encodings

Nowadays, modern telecommunication is mainly based on the following steps: voice information is digitalised (by sampling), then encoded, and then transmitted as a packets stream [4].

A. Encoding

The encoding process starts by producing digital "rough" results during the sampling phase, then normally reducing the bit rate (so, the bandwidth) of the sampled data through a suitable compression. There may be many coding techniques and among these we can mention:

- *Differences encodes*: if the next sample differs not much from the previous one, then we transmit the difference (which requires a lower number of bits) with respect to the original sample; a typical example is video encoding used in MPEG, which adopts a differential coding both regarding the previous frame and the next one [5].
- *Weighted encodes*: if certain samples are often present within the voice stream, we adopt a convention which codifies them through a smaller number of bits in order to save bandwidth (used e.g. in compression techniques such as ZIP) [5].
- *Loss encodes*: it is based on the principle that, for human ear, certain audio signals are practically ignored. This type of encoding causes such parts of signals to be erased, and the resulting encoding becomes leaner because there are less data to send (this technique is used in MP3 audio compression) [6].

With these techniques we can obtain significant signal compressions. Due to the various application fields, different types of codecs, characterised by different complexity, have been developed. In order to determine which of these should be used in a VoIP service, it is important to take into account the features in terms of bandwidth, encoding delay, and the voice quality reproduced on the receiving part. The main encoding used for the telephony transport on digital lines is the PCM, described in ITU-T G.711 recommendation, which produces a flow of 64 kbps [5]. This encoding is very simple and widespread (for the reasons mentioned above) particularly among telcos. The standard which offers the highest compression, maintaining good voice quality, is the DR SCS (Dual Rate Speech Coding Standard) or G.723. It can go up to speeds of 5.3 kbps, and is commonly adopted on videoconferencing systems over analog lines [5], [7].

Other popular voice coding standards for telephony and packet voice include [5]:

- G.711 - Describes the 64 Kbps PCM voice coding technique outlined earlier; G.711-encoded voice is already in the correct format for digital voice delivery in the public phone network or through Private Branch eXchanges (PBXs);
- G.728 - Describes a 16 Kbps low-delay variation of CELP voice compression;
- G.729 - Describes CELP compression that enables voice to be coded into 8 Kbps streams; two variations of this standard (G.729 and G.729 Annex A) differ largely in computational complexity, and both generally provide speech quality as good as that of 32 Kbps ADPCM.

B. Packets encoding

While the previous steps (sampling and coding) can be partially used even on a digital telephone network, the packaging operation is peculiar to packet networks. A package, by its nature, is composed by a series of headers so that the package can reach properly the destination. These headers can not be eliminated; this fact implies they must be present independently from the number of packets sent and their size. In voice over IP, the typical header has a size of 58 bytes (18 bytes Ethernet, 20 bytes IP, UDP 8 bytes, 12 bytes RTP): if each package carried only one data byte, the efficiency would become equal to approximately 1.7%, as a voice stream at 64kbps would generate a traffic of 3.7Mbps. Unfortunately, it is not possible to use packages of arbitrary size, because by decreasing the package encoding time, the delay would increase. A reasonable packaging delay values are in the order of 20-40 ms [5].

Compression efficiency is possible, in this scenario, to work around the problems as well as with respect to the used bandwidth, if the compression is achieved by other coding techniques, such as wavelet compression.

III. Wavelets

In recent years, wavelet theory has been developed as a unifying framework for a large number of techniques for wave signal processing applications, such as multiresolution analysis, sub-band coding and wavelet series expansions [8]. The idea of analysing a signal at various time frequency scales with different resolutions has emerged independently in many mathematics, physics and engineering fields. In fact wavelet analysis is capable of revealing aspects of data that other signal analysis techniques cannot take into account, especially when breakdown points, discontinuities in higher derivatives, and other self-similarity occur. Furthermore, because it let us obtain a different representation of data than those offered by traditional techniques, it can help us to efficiently compress or de-noise a signal without any appreciable degradation [9], [10], [11]. A significant advantage of using wavelets for speech coding is that the compression ratio can easily be optimised, while most other techniques have fixed compression ratios keeping all the other parameters constant.

Wavelet analysis is the breaking up of a signal into a set of scaled and translated versions of an original wavelet. The wavelet transform of a signal decomposes the original signal into wavelets coefficients at different scales and positions.

A. Wavelets in continuos domain

Wavelets are continuos "basis" functions constructed in order to satisfy certain mathematical properties. A wavelet is defined as a function $\psi(x)$ which must be subject to the following constrains:

1) $\int_{-\infty}^{+\infty} \psi(x)\, dx = 0$

2) $\| \psi(x) \|^2 = \int_{-\infty}^{+\infty} \psi(x)\psi^*(x)\, dx = 1$

Moreover, a whole family of wavelet functions can be obtained by just shifting and scaling as:

$$\psi_{j,k} = \sqrt{2^j}\ \psi(2^j t - k), \quad i, j \in \mathbb{N} \tag{1}$$

The idea behind wavelets is that by stretching and translating one such wavelet function $\psi(t)$ (also called mother wavelet), we can represent a signal $f(t) \in L^2(\mathbb{R})$ as:

$$f(t) = \sum_{j,k} b_{j,k}\ \psi_{j,k}(t) \tag{2}$$

where $b_{j,k}$ are called *wavelet coefficients* of the signal f in the wavelet basis given by the inner product of $\psi_{j,k}$ [12]. The wavelet coefficients represent the original signal in the wavelet domain [10]. An advantage of wavelet transforms is that the windows vary.

B. Discrete Time Case

In the discrete time case, two methods have been developed independently, namely *sub-band coding* (widely used in voice compression) and *multiresolution signal analysis*.

1) Multiresolution analysis: With this method we can derive a lower resolution signal by lowpass filtering with a half-band low-pass filtering having impulse response $g(n)$. This results in a signal $y(n)$ where

$$y(n) = \sum_{k=-\infty}^{k=+\infty} h(k) * f(2n - k) \tag{3}$$

Now based on subsampled version of $f(n)$ we want to find an approximation, $a(n)$, of the original signal $f(n)$: this is done by inserting a zero between every sample, because we need a signal at the original scale for comparison. Therefore, there is some redundancy in the number of samples, which will be proven useful for compression.

2) Sub-band coding schemes: The idea behind this technique is based on the following methodology: a pair of filters are used, i.e. a low pass and a high pass filter; we decompose a sequence $X(n)$ into two subsequences at half rate, or half resolution, and this by means of an orthogonal filter. This process can be iterated on either or both subsequences. In particular to obtain finer frequency resolution at lower frequencies, we iterate the scheme on the lower band only (see Figure 1). Thus, applying wavelet transform on a speech signal helps in compressing it to meet the stringent demands of bandwidth consumption while maintaining the quality and integrity of a signal [10].

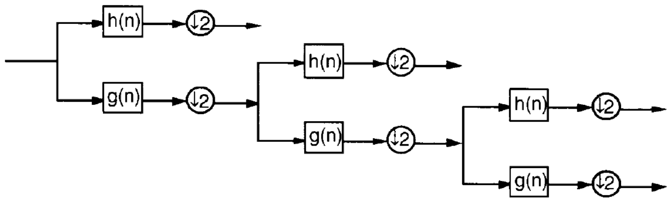


Fig. 1: Filter bank tree of discrete wavelet transform implemented with two discrete time filters (low-pass and high-pass filter, respectively)

C. Biorthogonality

The wavelet filter used in this work for audio signal compression is the 3.7 biorthogonal wavelet transform. A biorthogonal wavelet is a wavelet where the associated wavelet transform is invertible but not necessarily orthogonal; the first number indicates the order of the synthesis filter, while the second number refers to the order of the analysis filter. Even if frequency responses from biorthogonal filters may now not show any symmetry and the energy of the decomposed signal may also not equal the energy of the original signal, there are some reasons behind using biorthogonal wavelets, such as their compact support: this means we do not require IIR filters to approximate the signal. This fact is good because IIR filters are more difficult to treat and they also require extra computing power [12].

IV. GPU Parallel Computing with CUDA

The advent of multicore CPUs and many core GPUs have allowed the development of applications that transparently scale to take advantage of the increasing number of processor cores. Recently, the parallel programming field has seen an increasing success with the development of a novel technology, called Common Unified Device Architecture (CUDA).

A. GPU architecture

A CUDA-enabled GPU is composed of several MIMD (multiple instruction multiple data) multiprocessors that contain a set of SIMD (single instruction single data) processors. Each multiprocessor has a shared memory that can be accessed from each of its processors, and also shares a bigger global memory. Shared memory buffers reside physically on the GPU as opposed to residing in off-chip DRAM, so because of this, the latency to access shared memory tends to be far lower than typical buffers [13].

In CUDA programming model, an application consists of a *host* program that executes on the CPU and other parallel *kernel* programs executing on the GPU. A kernel program is executed by a set of parallel threads, where a thread is a subdivision of a process in two or more sub-processes, which are executed concurrently by a single processor. The host program can dynamically allocate device global memory to the GPU and copy data to/from such a memory from/to the memory on the CPU. Moreover, the host program determines

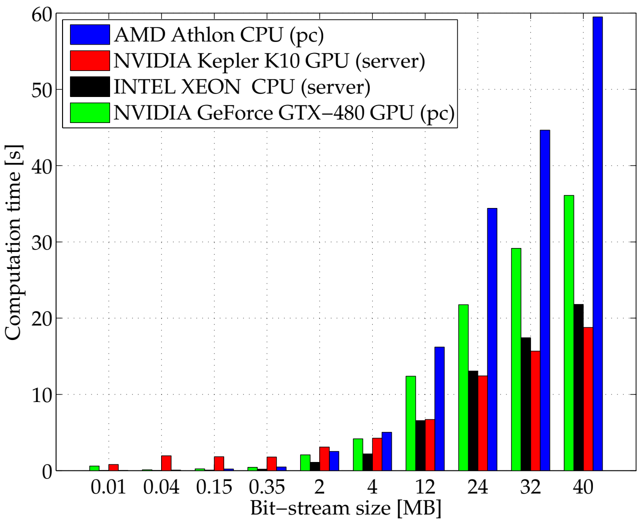


Fig. 2: Comparison among parallel and sequential programs on PC and Server

the number of threads used to execute kernel. Set of threads are grouped in blocks, and each block shares its own memory. It follows also that interactions between CPU and GPU should be minimised in order to avoid communication bottlenecks and delays due to data transfers [14], [15], [16].

B. *Performing wavelet transforms on GPU*

Classical wavelet transform implementations can be too computationally costly to cater for real time systems, hence we have investigated a parallel and fast approach to the problem at hand. As we have shown in [12], it is possible to adopt a fast GPU-oriented processing system to obtain the wavelet decomposition within the speed requirements of a VoIP service.

Figure 2 shows CPUs and GPUs timing performances for a conventional host, named PC, and for a Server. PC has been equipped with an AMD Athlon 64X2 having CPU clock frequency up to 2.9 GHz and a NVIDIA GeForce GTX 480 GPU; in Server, instead, has been equipped with an Intel Xeon CPU having a clock frequency up to 3.4 GHz, and a NVIDIA Tesla Kepler K10 GPU with 1536 cores.

V. Wavelet Compression

The goal of this work is to show the potential in terms of audio signals compression processed by wavelet filters. In particular, here we used a 3.7 biorthogonal wavelet filter. We will evaluate the wavelet compression efficiency and we will compare it with encoded audio trace through the typical PCM-16-bit VoIP standard at frequency of 8KHz. This analysis looks at both the quantitative aspects of encoding (the *compression factor*), and the audio quality perceived by listeners.

A. *Experimental Setup*

Initially an audio signal has been imported as a simple numerical array, whose size and the sampling frequency are fixed. In our setup, the audio signal has a duration of about 8 seconds and is sampled at a frequency of 22050 Hz. As explained in Section III, we know that, when we consider details with increasingly sample rate, the wavelet filter selects audio signal components which are at higher frequencies and which are characterised by a short time decay. These characteristics are typical of noise and background, therefore removable from the audio signal. It is then convenient to intervene more strongly on this portion of the information when coming to lossy compression. This result is achieved by appropriately setting thresholds. The detailed coefficients are removed from the last two sub-bands.

The result of superimposition between original and compressed signal is shown in Figure 3b. Once the two last sub-bands are removed, the coefficients set is compressed, taking also advantage of the zero-coefficients redundancy. As explained in Section III-B2, the sub-band coding process performs a dyadic signal processing. In other words, by the first subsampling cycle, we get a details vector, called *d1*, whose size is equal to half of original signal's size; for the second scale, a details vector *d2* is achieved, whose size will be equal to half of *d1* vector's size, i.e. one quarter (25 %) of the original signal size. Moreover, since we have suppressed the last two bands of the original signal, we take into account only the first quarter of the coefficient vector, because all other elements are consecutive zeros that were generated during the compression phase. Furthermore, the existence of other non-consecutive zeros inside the coefficient vector helps the compression further. Finally, we have compressed the coefficients using a gzip compressor. This final result is then transmitted by the VoIP client.

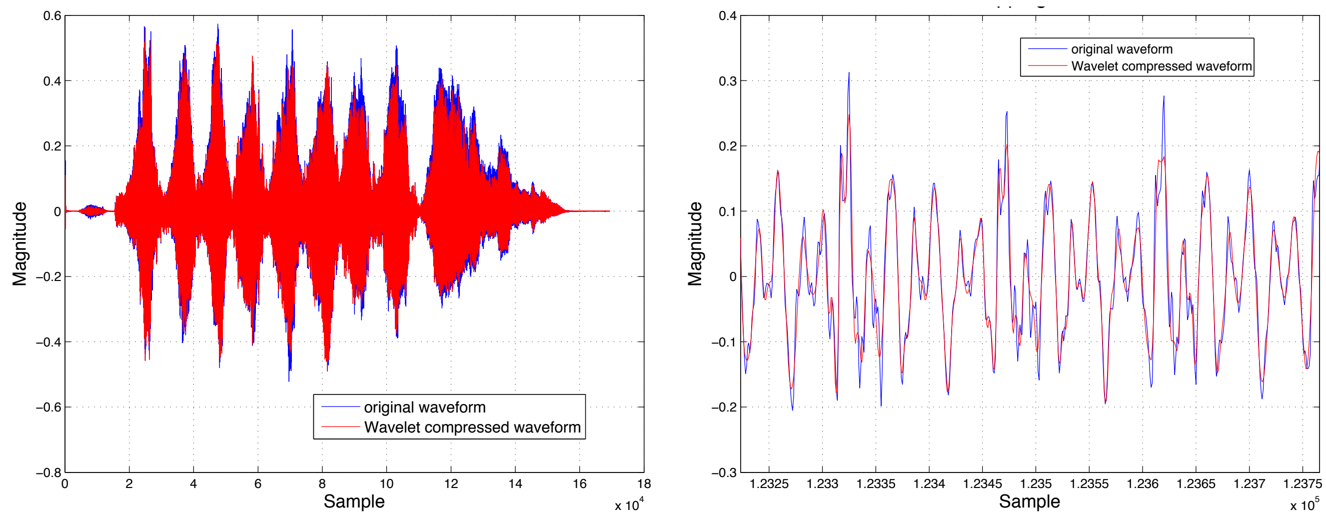
The reconstruction phase proceeds reversely to the compression stage: the starting point is the compressed data; the data stream is then decompressed with gzip. The decompressed data are initially incomplete, because we have to add zeros in order to reach the original size of the coefficients vector. In this way we can reconstruct the original digital signal.

B. *Streams*

The presented procedure is recursively applied for audio results. In this paper, we have analysed audio tracks that have a minimum duration of 8 seconds and a maximum of 10 minutes. For each of them, we have taken into account:

- the size of starting file, saved in .wav format and sampled at 22050 Hz;
- the size of file encoded using the 16-bit PCM to 8 KHz VoIP standard;
- the size of gzip-compressed file after wavelet filtering.

From the comparison among these three quantities, for different audio traces durations, the final result is shown in Figure 4. By observing Figure 4, it is possible to note a marked difference in the occupied bandwidth between the first two audio signals, namely the original and the VoIP one, compared to the proposed results obtained by wavelet compression. We have determined that the VoIP standard can give a compression factor of about 4, while the wavelet filtering allows us to obtain



(a) Global view of superimposition between original and compressed signal (b) Superimposition between original and compressed signal in detail

Fig. 3: Superimposition between original and wavelet compressed signal

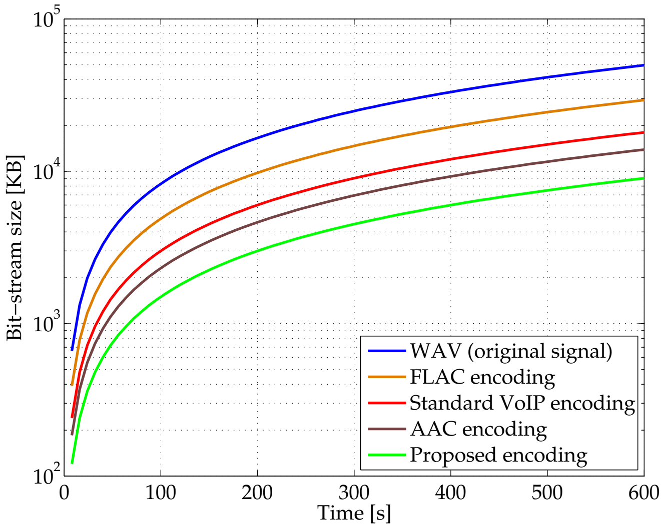


Fig. 4: Comparison among different standard audio data encodings with respect to our proposed wavelet compressed format

a compressed file approximately 10 times smaller than the original one. Therefore, our approach can improve the standard VoIP protocol compression.

VI. Related Works

Some works in the literature have analysed similar problems regarding audio compression techniques, and the relative loss of quality as well as regarding the delivery of content with limited network resources[17]. In [18] the authors use optimal adaptive wavelet selection and wavelet coefficients quantization procedures together with a dynamic dictionary approach. It takes advantages of the masking effect in human hearing. They minimise the number of hits required to represent each frame of audio material at a fixed distortion level. In [19], the authors show that the WP decomposition provides sufficient resolution to extract the time-frequency characteristics of the input signal and the WP based audio compressor provides transparent sound quality at compression rates comparable to the MPEG compressor with less than one third of the computational effort.

In [20], the authors have proposed a novel filter bank scheme which switches between a MDCT and a wavelet filter bank based on signal characteristics. A tree structured wavelet filter bank with properly designed filters offers natural advantages for the representation of non-stationary segments, such as attacks.

Interesting results are usually achieved when using a neural network approach to retain the relevant information on huge amount of sparse data [21], [22], [23], and some experiments have been performed on the compression of images [24]. Moreover, several solutions have been proposed for separating the components representing neural networks and the logic using them [25], [26], [27], [28]. In [29], the authors have proposed to apply wavelet analysis and audio compress technology in audio watermarking. Their approach performs a wavelet transform to determine the local audio properties. In [30] the authors have provided a novel scheme to join the wavelet packets and perceptual coding to construct an algorithm that is well suited to high-quality audio transfer for internet and storage applications. Finally, in [31] the authors examined how a compression method employs an approximation of a psychoacoustic model for wavelet packet decomposition. It has a bit rate control feedback loop particularly well suited to matching the output bit rate of the data compressor to the bandwidth capacity of a communication channel.

VII. Conclusion

In this paper we have investigated the compression potential of a 3.7 biorthogonal wavelet filter based technique for VoIP telecommunication. We have seen how, by using this compression technique, it is possible to obtain a greater compression factor than that obtainable with other traditional encodings. This fact is advantageous because the obtained audio stream is more efficiently compressed and requires less bandwidth for transmission.

The wavelet filtering is also interesting because makes it possible to implement it in a physical device. In fact, the proposed approach can be realised in hardware, basing on digital signals processing and compressing through wavelet filters. Furthermore we have to consider that nowadays modern 3G technology, widely used in mobile telephony, provides users with a quite limited data transmission bandwidth. For this reason wavelet compression approach, featured by a compression factor greater than ones used in other codings, meets practical needs of people who want to exchange data in a wireless medium.

Acknowledgements

This work has been supported by Project Knowledge Education Development financed by the European Social Fund POWR.03.03.00-00-P001/15.

References

[1] D. Monro, "Audio compression," August 2004, US Patent App. 10/473,649.

[2] O. O. Khalifa, S. H. Harding, and A.-H. Abdalla Hashim, "Compression using wavelet transform," *International Journal of Signal Processing*, vol. 2, no. 5, pp. 17–26, 2008.

[3] H. Kaur and R. Kaur, "Speech compression and decompression using DWT and DCT," *Int. J. Computer Technology and Applications*, vol. 3, no. 4, pp. 1501–1503, August 2012.

[4] F. D. Rango, M. Tropea, P. Fazio, and S. Marano, "Overview on VoIP: Subjective and Objective Measurement Methods," *International Journal of Computer Science and Network Security*, vol. 6, no. 1, pp. 140–153, 2006.

[5] J. Davidson, *Voice over IP fundamentals*. Cisco press, 2006.

[6] [Online]. Available: http://www.ucci.it/docs/ICTSecurity-2003-18b

[7] M. Mehić, M. Mikulec, M. Voznak, and L. Kapicak, "Creating covert channel using sip," in *Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2014, pp. 182–192.

[8] S. Mallat, *A wavelet tour of signal processing: the sparse way*. Academic press, 2008.

[9] D. L. Donoho, "De-noising by soft-thresholding," *IEEE Transactions on Information Theory*, vol. 41, no. 3, pp. 613–627, 1995.

[10] V. Malik, P. Singh, A. kumar Singh, and M. Singh, "Comparative analysis of Speech Compression on 8-bit and 16-bit data using different wavelets," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 4, no. 5, May 2013.

[11] M. B. Abdallah, J. Malek, A. T. Azar, H. Belmabrouk, J. E. Monreal, and K. Krissian, "Adaptive noise-reducing anisotropic diffusion filter," in *Federated Conference on Computer Science and Information Systems (FedCSIS)*. IEEE, 2015, pp. 1–28.

[12] A. Scivoletto and N. Romano, "Performances of a parallel cuda program for a biorthogonal wavelet filter," in *Proceedings of the International Symposium for Young Scientists in Technology, Engineering and Mathematics (System)*, 2016.

[13] J. Sanders and E. Kandrot, *CUDA BY EXAMPLE, An Introduction to General-Purpose GPU Programming, NVIDIA*.

[14] F. Bonanno, G. Capizzi, S. Coco, C. Napoli, A. Laudani, and G. Lo Sciuto, "Optimal thicknesses determination in a multilayer structure to improve the spp efficiency for photovoltaic devices by an hybrid femcascade neural network based approach," in *International Symposium on Power Electronics, Electrical Drives, Automation and Motion (SPEEDAM)*. IEEE, 2014, pp. 355–362.

[15] M. Wozniak, D. Polap, G. Borowik, and C. Napoli, "A first attempt to cloud-based user verification in distributed system," in *Asia-Pacific Conference on Computer Aided System Engineering (APCASE)*. IEEE, 2015, pp. 226–231.

[16] Z. Marszalek, M. Wozniak, G. Borowik, R. Wazirali, C. Napoli, G. Pappalardo, and E. Tramontana, "Benchmark tests on improved merge for big data processing," in *Computer Aided System Engineering (APCASE), 2015 Asia-Pacific Conference on*. IEEE, 2015, pp. 96–101.

[17] K. Kaczmarski, M. Pilarski, B. Banasiak, and C. Kabut, "Content delivery network monitoring with limited resources," in *Federated Conference on Computer Science and Information Systems (FedCSIS), 2013*. IEEE, 2013, pp. 801–805.

[18] D. Sinha and A. H. Tewfik, "Low Bit Rate Transparent Audio Compression using Adapted Wavelets," *IEEE Transactions On Signal Processing*, vol. 41, no. 12, December 1993.

[19] M. Black and M. Zeytinoglu, "Computationally efficient wavelet packet coding of wide-band stereo audio signals," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5. IEEE, 1995, pp. 3075–3078.

[20] D. Sinha and J. D. Johnston, "Audio compression at low bit rates using a signal adaptive switched filterbank," in *Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2. IEEE, 1996, pp. 1053–1056.

[21] C. Napoli, G. Pappalardo, M. Tina, and E. Tramontana, "Cooperative strategy for optimal management of smart grids by wavelet rnns and cloud computing," *IEEE Transactions on Neural Networks and Learning Systems*, (in press) 2015.

[22] C. Napoli, G. Pappalardo, E. Tramontana, R. Nowicki, J. Starczewski, and M. Woźniak, "Toward work groups classification based on probabilistic neural network approach," in *Proceedings of Artificial Intelligence and Soft Computing*, ser. Lecture Notes in Computer Science, vol. 9119. Springer, 2015, pp. 79–89.

[23] C. Napoli, G. Pappalardo, and E. Tramontana, "A mathematical model for file fragment diffusion and a neural predictor to manage priority queues over bittorrent," *International Journal of Applied Mathematics and Computer Science*, vol. 26, no. 1, pp. 147–160, 2016.

[24] M. Wozniak, C. Napoli, E. Tramontana, and G. Capizzi, "A multiscale image compressor with rbfnn and discrete wavelet decomposition," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2015, pp. 1219–1225.

[25] C. Napoli and E. Tramontana, "An object-oriented neural network toolbox based on design patterns," in *Proceedings of the International Conference on Information and Software Technologies (ICIST)*, ser. CCIS, vol. 538. Springer, 2015, pp. 388–399.

[26] A. Calvagna and E. Tramontana, "Delivering dependable reusable components by expressing and enforcing design decisions," in *Proceedings of IEEE Computer Software and Applications Conference (COMPSAC) Workshop QUORS*, Kyoto, Japan, July 2013, pp. 493–498.

[27] G. Capizzi, G. L. Sciuto, C. Napoli, E. Tramontana, and M. Woźniak, "A novel neural networks-based texture image processing algorithm for orange defects classification," *International Journal of Computer Science & Applications*, vol. 13, no. 2, pp. 45–60, 2016.

[28] R. Giunta, G. Pappalardo, and E. Tramontana, "Superimposing roles for design patterns into application classes by means of aspects," in *Proceedings of ACM Symposium on Applied Computing (SAC)*. Riva del Garda, Italy: ACM, March 2012, pp. 1866–1868.

[29] L. Cui, S.-x. Wang, and T. Sun, "The application of wavelet analysis and audio compression technology in digital audio watermarking," in *Proceedings of International Conference on Neural Networks and Signal Processing*, vol. 2. IEEE, 2003, pp. 1533–1537.

[30] P. Srinivasan and L. H. Jamieson, "High-Quality Audio Compression Using an Adaptive Wavelet Packet Decomposition and Psychoacoustic Modeling," *IEEE Transactions On Signal Processing*, vol. 46, no. 4, April 1998.

[31] K. Dobson, N. Whitney, K. Smart, P. Rigstad, J. Yang *et al.*, "Method and apparatus for wavelet based data compression having adaptive bit rate control for compression of digital audio or other sensory data," oct 1998, US Patent 5,819,215.