STATE OF THE ART AND NEW RESULTS IN DIRECT MANIPULATION OF MPEG AUDIO CODES

Goffredo Haus *Giancarlo Vercellesi*

Laboratorio di Informatica Musicale (LIM)
Dipartimento di Informatica e COmunicazione (DICO)
Università degli Studi di Milano
Via Comelico, 39 - Milano (Italia)
{haus,vercellesi}@dico.unimi.it

ABSTRACT

In this paper we present several approaches to direct manipulation of MPEG[1] audio codes. We discuss problems and solutions regarding aliases introduced by time to frequency transform block and NMR[2] modification. We also discuss the limits in term of processing functions and give some consideration about computational costs.

1. INTRODUCTION

The traditional approach to the manipulation of compressed audio codes consists in decoding the audio signal to PCM[3] and then, after manipulation, re-encoding it in compressed domain (**Figure 1**)

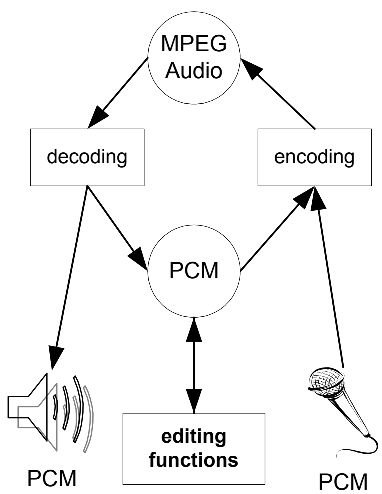

Figure 1: traditional manipulation of compressed audio formats

This approach has two disadvantages:

1. quality degradation as a result of tandem coding [13];
2. time and space consumption.

Furthermore, if we want to produce hardware chips that provide processing functionalities on compressed audio signal, whole codec must be built, which increases production costs.

Compressed domain processing allows editing audio signals without performing the whole decoding and re-encoding procedure (**Figure 2**). In this way it reduces time [14] and space requirements, while maintaining the perceived audio quality. It also reduces cost in manufacturing MP3 processing hardware. Encoder and decoder remain necessary only in order to play and record sounds.

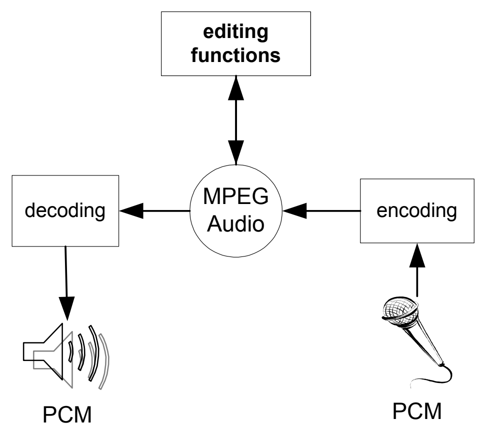

Figure 2: direct manipulation of compressed audio formats

This paper is organized as follows: in section 2 we revisit some basic concepts of MPEG standard, useful for a better comprehension of the rest of the work. In section 3 we describe problems there are present in direct manipulation of MPEG audio spectrum. In section 4, we illustrate different approaches in direct editing; in section 5 we present our algorithms to manipulate MP3 codes in quantized and Huffman domain. Finally, in section 6 conclusion and future works are illustrated.

2. OVERVIEW OF MPEG AUDIO STANDARD

In order to better understand the rest of the paper, a brief overview of the basic concepts about MPEG audio standard is provided.

---

[1] Motion Picture Expert Group
[2] Noise to Mask Ratio
[3] Pulse Code Modulation

MPEG audio provides a set of standards to lossy audio compression. Algorithms are classified by layers (1, 2 and 3) sorted by complexity and efficiency. They are included both in MPEG-1 [1] and MPEG-2 [2] that allow to work, respectively, on high (32, 44.1, 48 KHz) and low sampling frequencies (16, 22.05, 24 KHz). Additionally, there is another unofficial MPEG version called MPEG-2.5 which works on very low sampling frequencies (8, 11.025, 12 KHz).

MPEG audio codecs generally use a non-uniform quantization on frequency domain driven by perceptual model to compress PCM audio signal into a standard bitstream at various bitrate values (**Figure 3**).

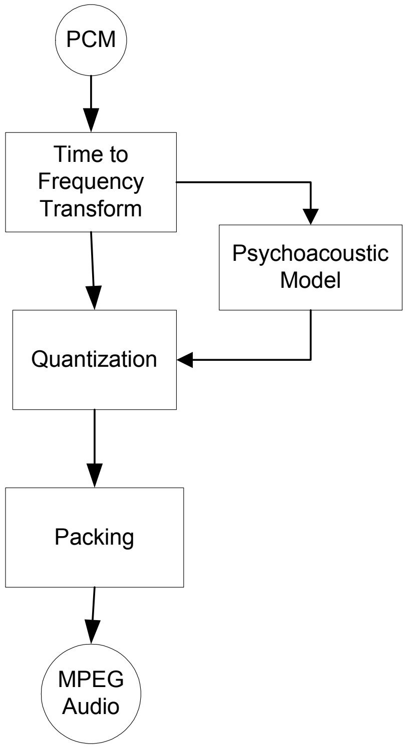

Figure 3: structure of MPEG codec

The time to frequency transform is built by means of a polyphase filter bank and, only in MP3 codecs, by cascading it with MDCT[1] (hybrid filter bank). Polyphase filter bank gets samples from PCM streaming and represents them in 32 frequency subbands, further subdivided into 18 finer subbands by MDCT in MP3.

Psychoacoustic model provides as its output the SMR[2], an index that informs the quantization block how many bits should be allocated for each frequency subband in order to get an inaudible quantization noise.

The output of filter banks and perceptual model are the input of non-uniform quantization process that decides how to quantize every frequency subband with respect to SMR value. Finally, MPEG audio bitstream is generated. In MP3 codecs, Huffman lossless compression is performed before bitstream packing. Further information about MPEG standards can be found in [3].

3. CONSIDERATION ON FILTER BANK ALIAS AND NMR IN SPECTRUM PROCESSING

In order to correctly process the spectrum we need to consider two important aspects: the frequency aliasing introduced by filter banks and the NMR information.

The analysis filter bank introduces alias that is partially removed in synthesis phase. In **Figure 4** we can see the behavior of aliasing during the various phases of MP3 encoding.

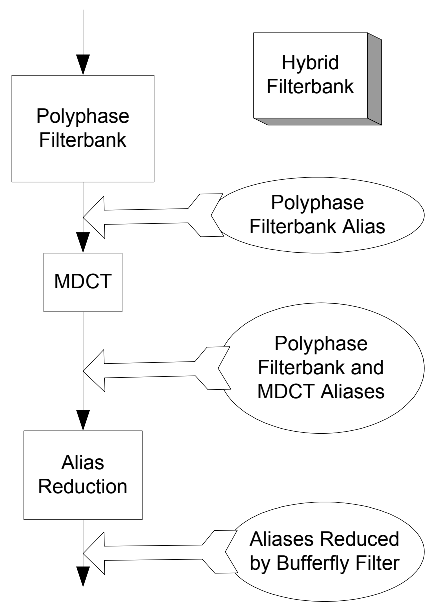

Figure 4: behavior of aliases during MP3 encoding; in MP1 and MP2, we have not MDCT and alias reduction blocks;

The hybrid filter bank introduces two different aliases, one by polyphase filter bank and one by MDCT. A butterfly filter further reduces these aliases in order to optimize the quantization phase. In decoding phase, only the MDCT alias is completely eliminated [4], while the polyphase alias is only reduced [5]. In MP1 and MP2 we only have the polyphase filter bank and alias reduction block is not present.

Every time we manipulate spectrum in compressed domain we introduce a modification of the alias shape such that a successful cancellation in synthesis phase would not be possible. This fact may introduce some audio artifacts, which would reduce the quality of perceived decoded audio signal. In order to avoid this, we must pay attention on alias shape during subbands coding.

---
[1] Modified Discrete Cosine Transform
[2] Signal to Mask Ratio

During spectrum processing, we usually change NMR information having masking or unmasking of components in audio signal [15]; this means that in some subband frequency the quantization noise, masked before processing, may became audible. So we have to care about quantization noise and masking structure during processing in subband coding keeping noise under audible threshold.

Various methods are developed by us and by other researches, in order to take care of these problems. They will be illustrated in next section.

4. DIFFERENT APPROACHES IN COMPRESSED DOMAIN EDITING

Generally, when we consider direct processing in compressed domain, we could make an attempt to manipulate the information directly, in their original structure. Unfortunately this is not always feasible. For example, it is not feasible to directly handle frequencies in MP3 codes because they are hidden within Huffman codes.

So, if we consider every block performed in MPEG representation to another, we can focus our attention on different parts of decoding phase. In this way we can get more approaches in direct editing with their own advantages or disadvantages, with respect to computational costs, perceived quality and processing limits.

In our work we have identified three relevant blocks in decoder. This allows us to formalize as many approaches in direct editing as possible (**Figure 5**): unpacking, Huffman decoding and dequantization.

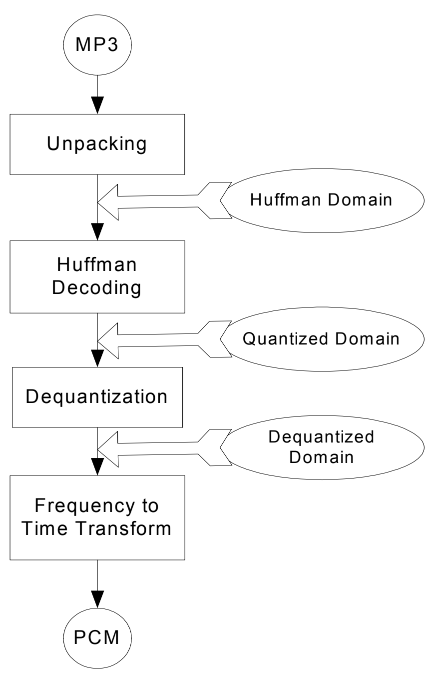

Figure 5: different approaches in direct manipulation of MP3

Keeping our attention to the output of these blocks we can imagine working on three different domains that we will call Huffman, quantized and dequantized. We can consider quantized and Huffman domain as the same in MPEG Layer 1 and 2 format, because they do not use Huffman algorithm.

We can just conclude that coming closer to the starting phase of the decoding process means to have a low computational cost in term of processing but it also means to have limits on the possible editing functions we may implement. On the other hand, as we approach the final phases of the decoding, we increase the possibility to develop more editing functions with better results, but it means also that we increase the computational cost of processing. Different application areas may privilege computational cost at a loss of quality of processing result, or vice versa.

In order to explain differences among various solutions, these approaches will be described in next sub-sections, showing for each example of processing function, alias and NMR the related problems and, when possible, the solutions. Finally, we shall give some considerations on computational costs.

4.1. Manipulation Performed After Dequantization

In this approach we deal with audio signals obtained from the output of dequantization phase (**Figure 6**) which represents the input of IMDCT[1] or polyphase filter bank, respectively in MP3 or MP1-MP2 cases. In MP3 codes we work on 576 dequantized frequency values (one granule) corresponded to 576 PCM samples; in MP2 codes we always have 576 values corresponded to as many as PCM samples but they are structured as matrix of 32 subbands with time window size of 18 samples. Finally, in MP1 we work on 384 samples structured as matrix of 32 subbands with time window size of 12 samples [1] [2].

As we said before, this approach allows developing nearly every kind of processing function getting results quite similar to the same functions performed in time domain, thanks to more possibility on managing alias shape and NMR. But on the other hand, despite of approaches in quantized and Huffman domain, this way is computationally heavier because it needs to perform a requantization driven by new SMR provided by psychoacoustic model (**Figure 6**).

On MP1 and MP2 format much research has been done. The previous work covers algorithms of gain control, mixing, subband filters, HRTF[2] spazialization [11] [12] and bitrate scaling [10], handling alias and NMR.

In [6] a first example of gain control, mixing and equalization is provided. In this work alias problem has been solved by developing FIR[3] filters which work independently on every polyphase subband where the

[1] Inverse Modified Cosine Transform

[2] Head-Related Transfer Function

[3] Finite Impulse Response

amplitude answer has a response magnitude quite similar among adjacent subbands. The requantization is performed by a convenient approximation of bit allocation.

In [7] an example of subband filter which avoids alias modification is provided. It generates FIR filters converting time domain filter to compressed domain filters, taking in consideration analysis and synthesis filter banks in a convenient manner. This way allows creating desired results in frequency domain without introducing distortions. The requantization phase is performed by estimation of new SMR from manipulated spectrum as illustred in [8].

In [9] another example of subband filter is provided. It follows a similar approach to the previous, but examples of IIR[1] filters are provided.

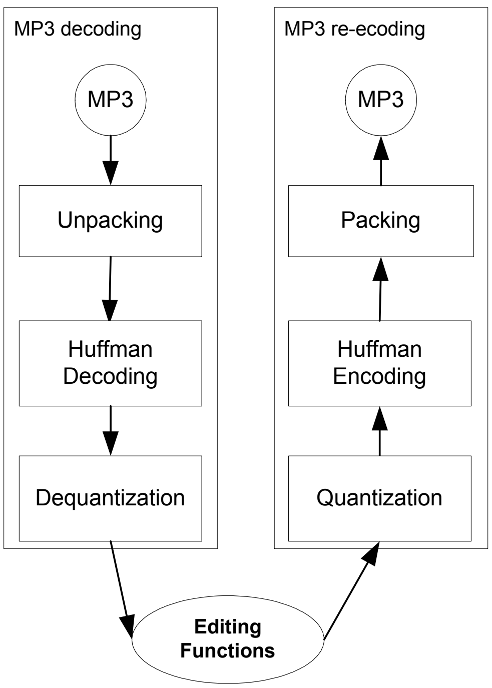

Figure 6: direct manipulation performed after dequantization

We believe that an approach similar to the one suggested by [7], [8] and [9] may be followed in order to develop subband filters and other editing functions in dequantized domain.

4.2. Manipulation Performed Before Dequantization

In this approach we deal with audio signals provided by Huffman decoder in MP3 format, or by unpacking in MP1 and MP2 format; both of them represent the input of requantization block (**Figure 7**).

Here we can directly manipulate the spectrum; its structure is close to dequantized domain but the frequency values are conveniently quantized by *scale factors*. Further, in MP3 codes, other parameters as *global gain* and subblock gain are used, and *scale factors* group frequencies in order to simulate critical bands.

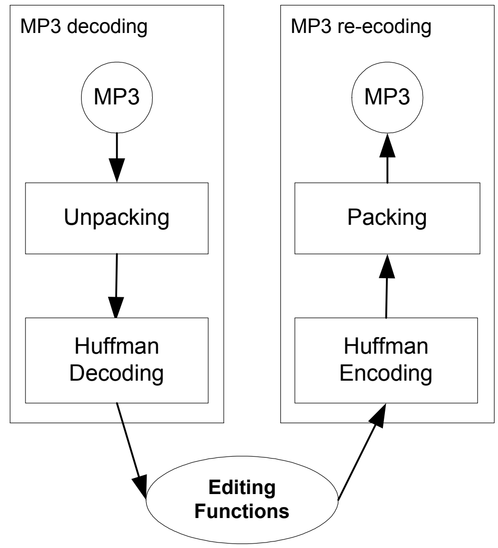

Figure 7: direct manipulation performed before dequantization

This approach represents a good compromise between computational costs and the number of implemented processing functions. With respect to the MPEG codec we only need to use packing/unpacking block. In the case of MP3 we use Huffman block, avoiding quantization phase, psychoacoustic model and, of course, hybrid filter banks. In this way we reduce the time of decoding and re-encoding. On the other hand, it is difficult to manage alias introduced by hybrid filter banks and NMR behavior. For this reason we can implement only limited number of processing functions.

In order to modify MP3 audio signal by reducing the introduction of artifacts caused by alias distortion or unmasking frequency, we have tried to reduce to minimum the manipulation of spectrum. So, to apply simple filters we set to zero all the frequency values that belong to stop-band. In this way we reduce bitrate value (useful to downgrade bitrate values), we avoid possible unmasking situation but, at the moment, we could have again problems with alias.

No related works are provided for MP1 and MP2 codes.

4.3. Manipulation Performed Before Huffman

In this approach we can consider only MPEG Layer 3 codes. Here we work on audio signals given by unpacking block (**Figure 8**).

Here, we can not directly manipulate the spectrum because it is represented by Huffman codebits; we are only allowed to change fields of MP3 bitstream as

[1] Infinite Impulse Response

parameters of dequantization formula, bitrate or *MDB*[1] value, in order to get the desired audio process.

This approach represents the best solution in term of computational cost because we only need to use the packing/unpacking block of MP3 codec. Furthermore, we don't have any difficulties related to alias because we do not change the spectrum structure. However, we could have problems with NMR changing *scale factor* values, and unmasking the quantization noise into their own critical bands. The negative side of this approach regards the limited number of editing techniques we may implement because we are able to manage only a limited number of parameters of coded audio signal. In our work we have developed two processing functions to directly move MP3 frames and to control the gain of MP3 granules.

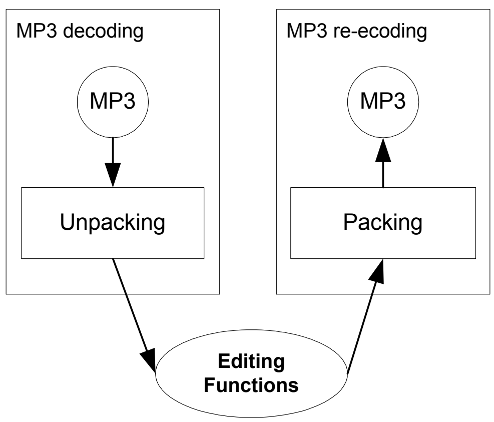

Figure 8: direct manipulation performed before Huffman decoding

Direct frame moving is not easy to realize in MP3 coding (in contrast to MP1 and MP2) because of the possible presence of bit reservoir (**Figure 10**) which creates a physical dependence between *main data* and their own frames. We eliminate this correlation by realigning *main data* into their own frames, conveniently changing bitrate values and *MDB*, and recreating a bitstream ISO-compatible. After this manipulation we get a new MP3 bitstream with average bitrate value a little greater than original.

In order to control the gain of MP3 codes we have chosen only to manage the *global gain* of dequantization formula avoiding modifying alias shape and NMR information, further maintaining bitrate value of processed signal audio equal to the original. These techniques will be illustrated in more detail in the next section.

5. ALGORITHMS TO EDIT MP3 CODES BEFORE DEQUANTIZATION AND HUFFMAN

In this section we illustrate techniques to move MP3 frames, control the gain and apply simple filters working in Huffman or quantized domain. We further give some observations on time resolution during direct manipulation of MP3 codes and re-encoding phase.

5.1. Time Resolution in Direct Manipulation of MP3

There are two approaches to time-domain processing of digital audio signals [16]:

1. sample-based and
2. block based.

For example, if we have to implement operations based on convolution formula we have to apply the block-based approach. Therefore we have to manage blocks of samples to perform processing operations; on the other side, cut & paste functions are generally sample-based so we have to handle directly the single PCM sample. It is easy to see that we have a better time resolution in sample by sample processing, equal to sampling time (1).

$$T_s = \frac{1}{F_s} \tag{1}$$

In contrast to time-domain, direct MP3 processing can be performed only by block-based, where their sizes are fixed by ISO standards [1] and [2. We can have two different atomic entities types, *frame* and *granule.* Frames have a time resolution equal to (2) where $N_{samples}$ = 1152 or 576, respectively with MPEG-1 or MPEG-2 / MPEG-2.5 standards. Granule has a time resolution provided by (2) where $N_{samples}$ = 576, both with MPEG-1 and MPEG-2 / MPEG-2.5.

$$T_S(N_{samples}) = \frac{N_{samples}}{F_S} \tag{2}$$

In cut & paste operation we work on frame, having time resolution equal to T_s (576 | 1152), depending on the standard used. In processing operations, as gain control or filters, we work on granule so we always have time resolution gave by T_s (576).

An example of MPEG-1 Layer 3 time structure with sampling frequency equal to 48 KHz is showed in **Figure 9**.

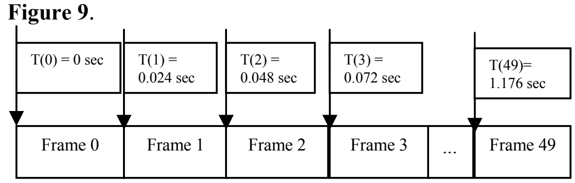

Figure 9: MPEG-1 Layer 3 time structure with sampling frequency 48 KHz

Here, we can see that each frame corresponds to 0.024 seconds (2). The time distance between the first and a generic frame F_k where K Є [0, Total Frames Number - 1] is given by:

$$TotTime(k) = k * T_s(1152) \tag{3}$$

---

[1] Main Data Begin

Finally, we can conclude that direct manipulation of MP3 codes generally has a lower editing resolution compared to the operations that can be done in the uncompressed domain. Further, time resolution is not dependent by bitrate.

5.2. MP3 Frames Moving

As we said before, it is difficult to move frames in MP3 because of the presence of bit reservoir technique. In this case, *main data* might not be completely contained into their frames but they might be spread over MP3 bitstream. Therefore we may have *main data* of frame F_{K+1} physically contained both on F_K and F_{K+1} (**Figure 10**:).

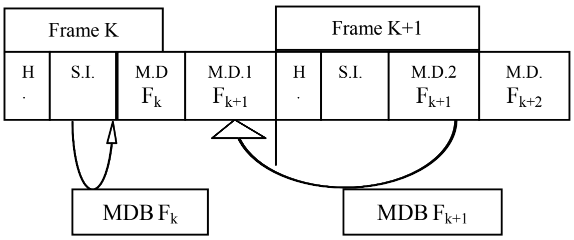

Figure 10: example of MP3 bitstream with technique of bit reservoir activated. We can see main data belong to F_{k+1} are contained in F_k

With this bitstream configuration we cannot move or delete frames without an appropriate alignment operation. Otherwise, we would introduce format incongruence and we would corrupt the audio information. An example of this corruption is showed in **Figure 11** where we deleted F_K. We can see how we have deleted part of *main data* belonged to F_K and how *MDB* refers to a wrong point in the MP3 bitstream.

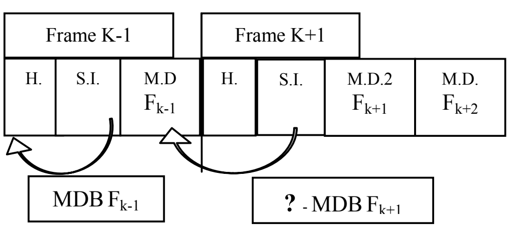

Figure 11: deletion of F_k in MP3 bitstream with bit reservoir activated without alignment technique. We can see MDB belongs to F_{k+1} points to incorrect place in MP3 bitstream. Furthermore, we have lost the first part of main data belong to F_{k+1}

To solve this problem we implemented an **alignment technique** in Huffman domain, which is able to put all *main data* into their own frames appropriately modifying bitrate and *MDB* values. Our alignment technique is subdivided in three phases; for each frame:

1. in the first phase, it reads and extracts information from header and side information fields and, through *MDB*, it finds the start of *main data* over MP3 bitstream. Then, through *part2_3_length* field [1], it calculates *main data* size and loads them into the main memory after its own side information. If ancillary data is present, our algorithm saves it in main memory and, if possible, rewrites it on disk during MP3 saving phase;
2. in the second phase, it deactivates bit reservoir technique setting to zero *MDB* field;
3. in the third phase (called **average bitrate)** it modifies bitrate values in order to have all *main data* contained in their frames. It calculates the size of MP3 frames by means of formulas provided by standards [1] [2]. A padding bit is used to round them when ever a decimal value is present. Then, it gets the real size of every frame as follow (4):

   $$Frs = Hs + CRCs + SIs + MDs \quad (4)$$

   Hs, CRCs and MDs are the byte size of header, CRC, side information and *main data* fields; MDs considers the reservoir pointed by *MDB* and ancillary data. At least, it finds the minimum bitrate value that provides a frame dimension by standard such as to contain their *main data*.

Generally, it is difficult to find bitrate values that provide perfect alignment between *main data* and frames. In the case we do not have a perfect realignment, we choose the first bitrate value (among default bitrate values provided by [1] and [2]) which corresponds to the frame size able to contain *main data*; extra bits will be processed by MP3 decoder as ancillary data. So, the average bitrate value of MP3 bitstream that results from the alignment technique can be obtained from the following formula (5):

$$orig\_br <= new\_br < NEXT\ (orig\_br) \quad (5)$$

orig_br and *new_br* are, respectively, the original and new bitrate value, and *NEXT (orig_br)* is a function whose output is the next bitrate value from the lists provided by [1] and [2]. For example, if a bitrate value is equal to 128 Kbit/sec (MPEG-1), then NEXT (128) = 160 Kbit/sec. It is important to note that it is possible to bring back *new_br* to *orig_br* applying again, on manipulated MP3 codes, the bit reservoir technique during re-encoding phase.

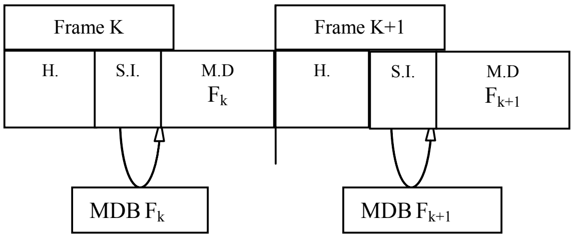

Figure 12: example of MP3 bitstream got after alignment technique. We can see every frame contains its own main data

After alignment technique has been applied, the frames are truly independent with one another and can be handled directly (**Figure 12**). So it is possible to cut/copy and paste, delete or invert frames, simply moving them in order to get the desired sequence.

In **Figure 13**, we show an example of frame inversion between frame 2 and frame 5. It is important to note that we invert frames position, and not their content.

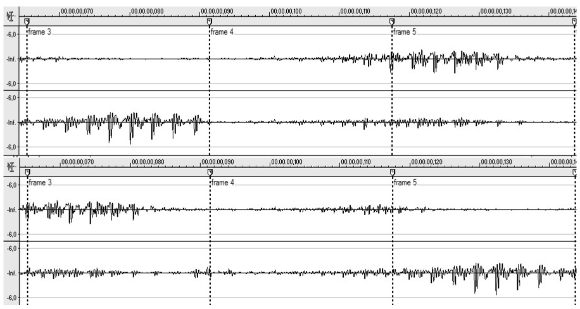

Figure 13: time-domain result got by frame inversion performed in compressed domain

Our technique is not able to align MP3 codes which are not strictly ISO compliant. This happens when *main data* size (*MDs*) is greater than the biggest dimension allowed at maximum bitrate value (table 1).

KHz *Byte*	*Fs ind.=0* *44,1|22,05|11,025* *Pad = 1*	*Fc ind.=1* *48|24|12*	*Fc ind.=2* *32|16|8*
MPEG-1	1045	960	1440
MPEG-2	523	480	720
MPEG-2.5	1045	960	1440

Table 1: the greatest *MDs* size at max bitrate value (320, 160, 160 Kbit / Sec) without considering header, CRC, side info and ancillary data sizes

5.3. Gain Control

Gain control is used to adjust the intensity of audio signals. For uncompressed codes, it is simply a multiplication of each PCM sample by a constant value. Instead, for MP3 codes we may take two different approaches:

1. direct manipulation of *global gain* field, operating in Huffman domain;
2. direct manipulation of 576 frequency values that belong to a granule, operating in dequantized domain.

We have chosen the first one in order to avoid the introduction of artifacts caused by aliases distortion or unmasking frequency. Furthermore, in contrast to the second approach, here we do not change the bitrate value.

During encoding quantization phase, if there is no valid Huffman table or if bitrate value is not respected, rate loop opportunely quantize spectrum reducing the values of 576 frequency lines increasing *global gain* field [1] [3]. Then, in decoding phase dequantization bring back spectrum in original as follow:

$$xr[i] = round\left[sign(xq[i]) * |xq[i]| * 2^{\left(\frac{GlobalGain - SYS}{4}\right)} \right] \quad (6)$$

where $i \in [0 : 575]$, xr[i] is the vector of unquantized frequency values, xq[i] is the vector of quantized frequency values and SYS is a system value which depends on the encoder implementation. A *global gain* equal to zero corresponds to no quantization, while 255 is the greatest value this field could assume [1] [2].

We can conclude from (6) that if we increase or decrease *global gain* without modifying frequencies, we increment or to decrement perceived intensity of decode MP3 with steps equal to 1.5 DB as showed in (7).

$$20 * Log_{10}\left(2^{\frac{1}{4}}\right) = 1{,}5\,DB \quad (7)$$

Therefore, increasing or decreasing *global gain* by *k* factor means to up or down RMS[1] by *k* * 1.5 Db (8). Our empiric experiments have shown the validity of this relationship.

$$global\_gain \pm k \equiv k * 1.5 \pm RMS \quad (8)$$

By changing the value *k* over the time, we can also obtain various dynamic effects like fade in, fade out or tremolo.

However, we can not always increment *global gain* because otherwise clipping might happen in time domain. A preemptive analysis may be useful in order to detect frames which could cause clip and, in these cases, we should modify *global gain* at will avoiding possible distortions. Our future work will be dedicated to this problem.

Finally, some MP3 frames may have *global gain* field set on greatest or smallest value and in these cases we cannot increment or decrement this parameter. For example, we cannot decrement the intensity of MP3 frame with *global gain* equal to 0 or, vice versa, increment the volume of MP3 frames with *global gain*

[1] Root Means Square

equal to 255. In these cases, our algorithm introduces no changes.

5.4. Filters

Filters and equalizations are typical operations which shape audio spectrum in order to give a desired frequency response to audio signal. When we work in time domain, we transform the signal in a frequency domain, we apply a desired transfer function and then we bring back the filtered signal in time domain. In MP3 codes we have just PCM audio signal transformed in frequency subbands coded so we only have to find a way to handle the quantized spectrum in order to apply filters or equalizations.

MDCT produces on output a granule which contains either 576 frequency lines, wrapped in a unique long block, or the same information further split in three short blocks of 192 frequency values. A mix of these two possibilities is also possible. The frequency resolution of our filter is equals to (9) with long blocks and (10) with short blocks.

$$F_{res}^{long} = round\left(\frac{F_s}{576}\right) \tag{9}$$

$$F_{res}^{short} = round\left(\frac{F_s}{192}\right) \tag{10}$$

If we have input filter with cut frequency not-multiple of F_{res}^{long} or F_{res}^{short}, we round it to the nearest value. For example, MP3 frames at 44.1 KHz have a frequency resolution equal to 76.5625 Hz for long block and 229.6875 Hz for short block; if we suppose to have a frame coded by long block and we want to apply it a low pass filter with cut frequency equal to 1000 Hz, our technique filter only F_i frequency values where i = 4.

As it happens also with gain control, even here there are two possible ways to filter audio signals before dequantization:

1. modifying *scale factors,* operating in Huffman domain
2. modifying MDCT frequencies, operating in quantized domain

In both we have problems with NMR and aliases because we may have to change the spectrum structure. Furthermore, only the second one allows reducing the bitrate value.

However, it is possible to avoid NMR problems and consequently, possible unmasking of quantization noises, following the second approach by setting to zero frequency values that belong to stop-band range [F_{start} : F_{stop}]. F_{start} and F_{stop} are calculated as shown before by (9) and (10). The other frequencies remain unchanged. In this manner, we also reduce bitrate value (11).

$$xq'[i] = round\ [sign\ (xq[i]) * |xq[i]| * R[i]]$$

where $R[i] = \begin{cases} 1 & if \quad i \in [F_{start} : F_{stop}] \\ 0 & \quad otherwise \end{cases} \tag{11}$

In this manner, we are able to provide all kinds of filters (low pass, high pass, band pass or band reject) simply giving in input to (11) a desired R[i] vector. But, these kind of filter sounds quite unnatural despite of being performed in dequantized domain because there are evident discontinuities between pass-band and stop-band edges as shown in **Figure 15** and **Figure 16**. **Figure 14** shows the original spectrum of a song; in **Figure 15** and **Figure 16** we have spectrum filtered, respectively, by band reject and band pass filters.

Then, some possible artifacts would be perceived after filters because, as explained in section 2, we could change aliases introduced by hybrid filter bank (and opportunely modified by alias reduction). This problem represents another chapter of our future work.

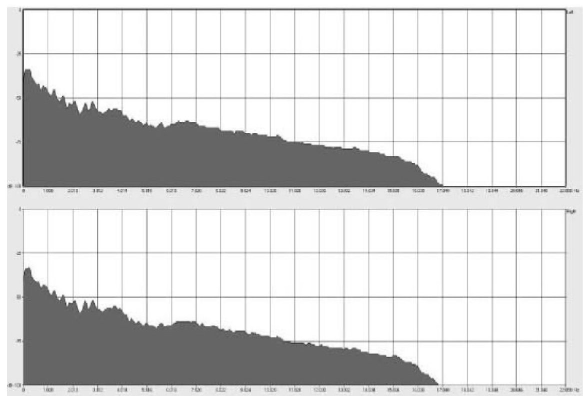

Figure 14: original spectrum of music audio signal

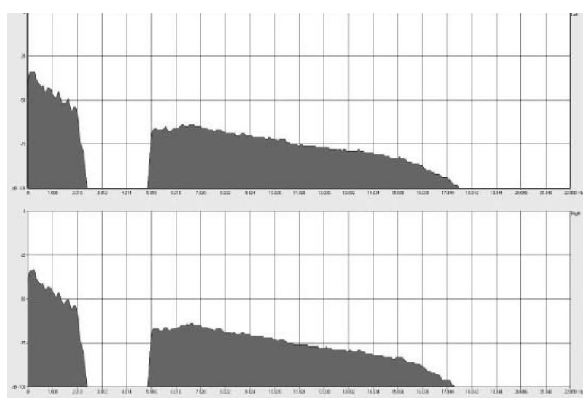

Figure 15: spectrum of music audio signal after application of band reject filter

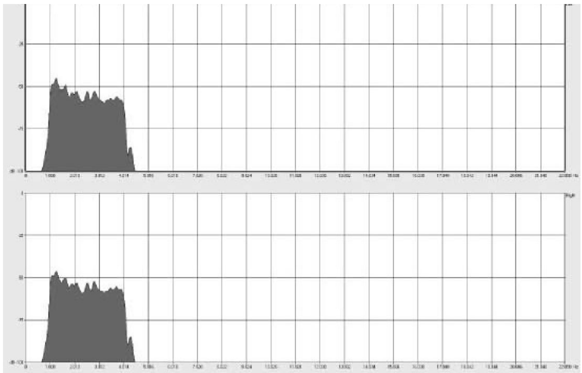

Figure 16: spectrum of music audio signal after application of band pass filter.

5.5. The Re-Encoding Process

After direct manipulation performed in quantized domain, it is necessary to perform a re-encoding in order to obtain a new MP3 bitstream. Staying just before dequantization block, we only have to execute Huffman encoding and packing block. This means that we have to recalculate new Huffman regions (finding correct tables to use) and then to create a new ISO compliant bitstream. In packing block, some aspects as bitrate, bit reservoir, CRC and windows switching pattern should be considered. In our work we do not modify windows switching pattern and we apply CRC, as explained by ISO standards, if it is requested by user.

As we said before, bitrate may change in different ways depending on techniques applied on MP3 bitstream: generally, it decreases with filters, stays the same with gain control and change as shown by (5) if we move the frames directly.

Finally, we are able to move frames and control the gain in both mono and stereo coding. Additionally, with separated channels it is possible to apply different kinds of filters for each channel, obtaining in this way interesting audio effects. At the moment, we are not able to apply filters in joint stereo coding.

6. CONCLUSION

In this paper we have formalized the kind of approaches in direct manipulation of MPEG codes and we have showed algorithms to manipulate MP3 codes directly. We saw that working before dequantization takes advantages in terms of computational costs but it introduces problems in alias and NMR managing, limiting the power of applied editing functions. So, future works are mainly addressed toward this direction, in order to find a way to avoid artifacts introduced by aliases modification after spectrum manipulation. Then, an estimation of SMR from current quantization may be useful to drive requantization of modified spectrum, working directly on parameters of dequantization formula. This allows us to increase or decrease frequency values as showed in [6] avoiding aliases artifacts and without generating perceptible quantization noise.

Other future works regarding dequantized domain are been scheduled. The aim is the develop subband filters and other digital audio processing and effects that can be applied after dequantization, following strategies showed in [7], [8] and [9]. In this case, it is necessary to implement transfer functions derived by time-domain filters and MP3 hybrid filter bank and a modified psychoacoustic model, which should be able to take in input dequantized MDCT coefficients instead of DFT[1].

7. ACKNOWLEDGMENTS

The authors wish to acknowledge the partial support of this project by Italian MIUR (FIRB “Web-Minds” project N. RBNE01WEJT_005) and the Italian National Re-search Council, in the framework of the research program “Methodologies, techniques, and computer tools for the preservation, the structural organization, and the intelligent query of musical audio archives stored on heterogeneous magnetic media”, Finalized Project “Cultural Heritage”, Subproject 3, Topic 3.2, Subtopic 3.2.2, Target 3.2.1.

This work has been made possible by the efforts of researchers and graduate students of LIM.

8. REFERENCES

[1] ISO/IEC International Standard IS 11172-3 "Information Technology - Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbits/s - Part 3: Audio"

[2] ISO/IEC International Standard IS 13818-3 "Information Technology - Generic Coding of Moving Pictures and Associated Audio, Part 3: Audio"

[3] D. Noll, “MPEG Digital Audio Coding”, IEEE Signal Processing Magazine, Vol. 14, No. 5, pp 59-81, 1997

[4] J. Princen, A. Johnson, A. Bradley, "Subband / Transform Coding Technique Based on Time Domain Aliasing Cancellation," Proc. of the Int. Conf. IEEE ICASSP, pp. 2161-2164, 1987

[5] J.H.Rothweiler, "Polyphase Quadrature Filters - a New Subband Coding Technique," Proc of the Int. Conf. IEEE ASSP, 27.2, pp 1280-1283, Boston, 1983

[6] M. A. Broadhead, Charles C. B. Owen, “Direct Manipulation of MPEG Compressed Digital Audio”, Proceedings of the third ACM International Conference on Multimedia, pp

[1] Discrete Fourier Transform

499-507, San Francisco, California, United States, 1995

[7] C. A. Lanciani, R. W. Schafer, “Subband-Domain Filtering of MPEG Audio Signals”, IEEE International Conference on Acoustic, Speech and Signal Processing, Arizona, USA, 1999

[8] C. A. Lanciani, R. W. Schafer, “Psychoacoustically - Based Processing of MPEG-1 Layer 1-2 Encoded Signal”, IEEE Workshop on Multimedia Signal Processing, pp. 53-58, 1997

[9] A. B. Benjelloun , “A Generic Framework for Filtering in Subband-Domain”, Proceeding of IEEE 9th Workshop on Digital Signal Processing, Hunt, Texas, USA, October 2000

[10] Y. Nakajiama, H. Yanagihara, A. Yoneyama, and M. Sugano, “MPEG Audio Bit Rate Scaling on Coded Data Domain”, IEEE, 1998

[11] A. B. Touimi, M. Emerit, J. Pernaux, "Efficient method for multiple compressed audio streams spatialization", Proceedings of the 3rd ACM International Conference on Mobile and Ubiquitous Multimedia, pp 229-235, College Park, Maryland, 2004

[12] C. A. Lanciani, R. W, Schafer, “Application of Head-Related Transfer Functions to MPEG Audio Signals”, Proceedings of IEEE International Conference on Acoustic, Speech and Signal Processing, 1999

[13] M. Erne, “Perceptual Audio Coders: What to Listen for”, Proceedings of 111th Convention of AES, Preprint 5489, New York, 2001

[14] M. M. Goodwin, M. Wolters, R. Sridharan, “Post-Processing and Computation in Parametric and Transform Audio Coders”, AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio, Espoo, 2002

[15] M. M. Goodwin, A.J. Hipple; B. Link, “Predicting and preventing unmasking incurred in coded audio post-processing”, IEEE Transactions on Speech and Audio Processing, Volume 13, Issue 1, Page(s):32 - 41, 2005

[16] Udo Zölzer, “DAFX - Digital Audio Effects”, John Wiley & Sons, 2002