REDUCED COMPLEXITY TRANSFORM FOR A LOW-FREQUENCY-EFFECTS CHANNEL

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority to United States Patent Provisional Application No. 61/468,373, filed 28 March 2011, hereby incorporated by reference in its entirety.

TECHNICAL FIELD

The present invention pertains generally to digital signal processing and is directed more particularly toward methods and apparatuses that may be used to apply filterbanks to limited bandwidth audio channels such as so called low-frequency-effects (LFE) channels using fewer computational resources.

BACKGROUND ART

Several international, regional and national standards have been developed to define methods and systems that may be used to implement multichannel audio coding systems. Three examples of such standards include ISO/IEC 13818-7, Advanced Audio Coding (AAC), also known as "MPEG-2 AAC," and ISO/IEC 14496-3, subpart 4, also known as "MPEG-4 audio," published by the International Standards Organization (ISO), and a standard published by the United States Advanced Television Systems Committee (ATSC), Inc. in Document A/52B entitled "Digital Audio Compression Standard (AC-3, E-AC-3)," Revision B, published June 14, 2005, also known as "Dolby Digital" or "AC-3."

Audio systems that conform to standards like those mentioned above generally include transmitters that apply an analysis filterbank to each of several channels of input audio signals, process the output of the analysis filterbanks into encoded signals and transmit or record the encoded signals, and receivers that receive the encoded signals, decode them and apply synthesis filterbanks to the decoded signals to generate channels of output audio signals that are a replica of the original input audio signals. Many of the standards specify implementing the analysis and synthesis filterbanks by a Modified Discrete Transform (MDCT) and an Inverse Modified Discrete Transform (IMDCT) described in Princen, Johnson, and Bradley, "Subband/Transform Coding Using Filter Bank Designs Based on Time Domain Aliasing Cancellation," *ICASSP 1987 Conf. Proc.*, May 1987, pp. 2161-64.

Filterbanks that are implemented by these particular transforms have many attractive properties but significant processing or computational resources are required to

perform the needed calculations. Techniques are known that can be used to perform the transforms more efficiently, thereby reducing the amount of computational resources that are needed. One characteristic that is common to these techniques is that their computational complexity varies with the so-called length of the transform. Techniques are known that can realize further reductions in computational complexity by using shorter transform lengths to process audio channels with narrower bandwidths.

Standards like those mentioned above define sequences of digital data or digital bit streams that carry data representing encoded representations of one or more audio channels. One configuration of channels sometimes referred to as "5.1 channels" includes five full-bandwidth channels denoted left (L), right (R), center (C), left-surround (LS), and right-surround (RS), and one limited-bandwidth channel or low-frequency-effects (LFE) channel. The full-bandwidth channels typically have a bandwidth of about 20 kHz and the limited-bandwidth LFE channel typically has a bandwidth of about 100 to 200 Hz. Because the bandwidth of the LFE channel is narrower, known techniques can be used to perform a filterbank transform more efficiently for the LFE channel than can be performed for one of the full-bandwidth channels.

Nevertheless, there is a need to develop techniques that further improve the efficiency of the transform filterbanks that are applied to limited-bandwidth channels like the LFE channel.

DISCLOSURE OF INVENTION

It is an object of the present invention to provide ways that can be used to perform transforms that implement filterbanks for limited-bandwidth channel signals more efficiently than is possible using known techniques.

According to one aspect of the present invention, a limited-bandwidth signal is processed by receiving a block of K real-valued transform coefficients of which only a number L of the coefficients represent spectral components of a limited-bandwidth audio signal, where $½\,L < M < K$, and M is a power of two; applying a first-transform of length R to a block of complex-valued coefficients derived from M complex-valued transform coefficients that include the L real-valued transform coefficients representing spectral components of the limited-bandwidth audio signal, wherein $R = \frac{M}{P}$ and P is a power of two; applying a bank of Q second transforms of length P to outputs of the first transform; and deriving a sequence of N real-valued signal samples from outputs of

the bank of second transforms, where $N = 2 \cdot K$ and the real-valued signal samples represent temporal components of the limited-bandwidth audio signal.

The various features of the present invention and its preferred embodiments may be better understood by referring to the following discussion and the accompanying drawings in which like reference numerals refer to like elements in the several figures. The contents of the following discussion and the drawings are set forth as examples only and should not be understood to represent limitations upon the scope of the present invention.

BRIEF DESCRIPTION OF DRAWINGS

Fig. 1 is a schematic block diagram of an audio coding system in which various aspects of the present invention may be carried out.

Fig. 2 is a schematic block diagram of a process that may be used to perform a synthesis transform in the coding system shown in Fig. 1.

Figs. 3 and 4 are schematic block diagrams that illustrate some features that may be used to perform a portion of the process shown in Fig. 2.

Fig. 5 is a schematic block diagram of a device that may be used to implement various aspects of the present invention.

MODES FOR CARRYING OUT THE INVENTION

A. Introduction

Fig. 1 is a schematic illustration of a two-channel audio coding system that includes a transmitter 100 and a receiver 200. The transmitter 100 receives from the paths 11, 12 two channels of input audio signals. The analysis filterbanks 111, 112 are applied to the input audio channels to obtain a first set of frequency-subband signals representing the spectral content of the input audio signals. These analysis filterbanks are implemented by time-domain to frequency-domain transforms. The encoder 120 applies an encoding process to the first set of frequency-subband signals to generate encoded information, which is passed along the path 20. The receiver 200 receives the encoded information from the path 20. Decoder 220 applies a decoding process to the encoded information to obtain a second set of frequency-subband signals. The synthesis filterbanks 231, 232 are applied to the second set of frequency-subband signals to generate two or more channels of output audio signals, which are passed along the paths 31, 32. These synthesis filterbanks are implemented by frequency-domain to time-domain transforms. The path

20 may be a broadcast medium, a point-to-point communication medium, a recording medium or any other medium that can convey or record the encoded information.

The encoder 120 and the decoder 220 are not essential to practice the present invention. If they are used, they may perform either lossless or lossy coding processes. The present invention is not limited to any particular encoding and decoding processes.

Only two channels of input and output audio signals are shown in the drawings for illustrative clarity. In many implementations there are more than two channels of input audio signals and more than two channels of output audio signals. At least one of the output audio signals has a bandwidth that is much narrower than the bandwidth of one or more of the other output audio signals.

The present invention is directed toward reducing the computational resources needed to perform the transform that implements the synthesis filterbank 231 or 232 in the receiver 200 used to generate narrower bandwidth output audio signals. The present invention can implement a more efficient synthesis filterbank in a receiver 200 that maintains compatibility with an analysis filterbank in existing transmitters 100.

The present invention may also be used to reduce the computational resources needed to perform the transform that implements the analysis filterbank 111 or 112 in the transmitter 100 applied to narrower bandwidth input audio signals. This implementation can maintain compatibility with a synthesis filterbank in existing receivers 200.

B. Implementation Techniques

Synthesis filterbanks can be implemented by a wide variety of frequency-domain to time-domain transforms including many variations of the Inverse Discrete Cosine Transform (IDCT) and the Inverse Modified Discrete Cosine Transform (IMDCT) mentioned above. The algorithms that define these transforms in a direct manner are referred to herein as "direct transforms."

One technique referred to herein as a "folding technique" can be used to perform these direct transforms more efficiently. The folding technique comprises three stages as illustrated in Fig. #2. The second stage 402 performs a transform that has a shorter length than the direct transform this folding technique implements. The transform that is performed in the second stage 402 is referred to as a "folded transform" so that the following description can more easily distinguish it from the direct transform.

The pre-processor stage 401 combines transform coefficients in a block of K real-valued frequency-domain transform coefficients into a block of $\frac{1}{2} \cdot K$ complex-valued transform coefficients. The transform stage 402 applies a frequency-domain to

time-domain folded transform of length $\frac{1}{2} \cdot K$ to the block of complex-valued transform coefficients to generate $\frac{1}{2} \cdot K$ complex-valued time-domain samples. The post-processor stage 403 derives a sequence of K real-valued time-domain samples from the $\frac{1}{2} \cdot K$ complex-valued time-domain signal samples. Except for any errors that might occur from finite-precision arithmetic operations, the K time-domain signal samples that are obtained by this technique are identical to the K time-domain signal samples that could be obtained by applying the direct transform of length K to the block of K real-valued frequency-domain transform coefficients. This technique improves efficiency because the additional computational resources needed to perform the direct transform as opposed to the folded transform in the stage 402 is greater than the computational resources needed to implement the processes performed in the pre-processor stage 401 and the post-processor stage 403.

If a block of transform coefficients represents a narrow-bandwidth signal in which a significant number of the transform coefficients are always zero, an additional transform-decomposition technique may be used to increase processing efficiency of the folded transform that is performed in the stage 402.

This technique is discussed in the following sections.

1. Direct Transforms

The direct IMDCT is shown in expression 2. Its complementary Modified Discrete Cosine Transform (MDCT) is shown in expression 1.

$$X(k) = \sum_{n=0}^{N-1} x(n) \cdot \cos\left[\frac{2\pi}{N}\left(n + \frac{1}{2} + \frac{N}{4}\right)\left(k + \frac{1}{2}\right)\right] \quad \text{for } 0 \le k < K \tag{1}$$

$$x(n) = \frac{2}{N} \sum_{k=0}^{K-1} X(k) \cdot \cos\left[\frac{2\pi}{N}\left(n + \frac{1}{2} + \frac{N}{4}\right)\left(k + \frac{1}{2}\right)\right] \quad \text{for } 0 \le n < N \tag{2}$$

where $X(k)$ = real-valued frequency-domain transform coefficient k;

K = total number of real-valued frequency-domain transform coefficients;

$x(n)$ = real-valued time-domain signal sample n; and

N = length of time-domain window of samples, where $N = 2K$.

The proper operation of these direct transforms requires the use of analysis window functions and synthesis window functions whose lengths and shapes meet certain requirements that are well known in the art. The analysis window function is applied to segments of N input audio signal samples prior to the application of the MDCT. The synthesis window function is applied to segments of N samples obtained from an

application of the IMDCT to blocks of K transform coefficients and these windowed segments of samples are overlapped and added to windowed segments of samples obtained from other blocks of transform coefficients. Additional details may be obtained from the Princen et al. paper cited above. The following paragraphs omit further discussion of the analysis window function.

2. Folding Technique

The process performed in the pre-processor stage 401 may be expressed as:

$$X'(k) = \left[X\left(\frac{N}{2} - 2k - 1\right) + j \cdot X(2k) \right] \cdot \left[-\cos\left(\frac{2\pi(8k+1)}{8N}\right) - j \cdot \sin\left(\frac{2\pi(8k+1)}{8N}\right) \right] \quad \text{for} \quad 0 \le k < \frac{N}{4} \tag{3}$$

where $X'(k)$=complex-valued frequency-domain transform coefficient k; and

j = imaginary operator equal to $\sqrt{-1}$.

The folded transform performed in the transform stage 402 may be expressed as:

$$x'(n) = \sum_{k=0}^{\frac{N}{4}-1} X'(k) \cdot \left(W_{N/4}\right)^{n \cdot k} \tag{4}$$

where $W_N = e^{j \cdot \left(\frac{2\pi}{N}\right)}$; and

$x'(n)$=complex-valued time-domain signal sample.

The process performed in the post-processor stage 403 may be expressed as:

$$y(n) = \left(\mathrm{Re}\left[x'(n)\right] + j \cdot \mathrm{Im}\left[x'(n)\right]\right) \cdot \left(-\cos\left(\frac{2\pi(8n+1)}{8N}\right) - j \cdot \sin\left(\frac{2\pi(8n+1)}{8N}\right)\right) \tag{5}$$

where $y(n)$ = intermediate sample value used in subsequent windowing calculations;

$\mathrm{Re}\left[x'(n)\right]$ = real part of the complex value $x'(n)$; and

$\mathrm{Im}\left[x'(n)\right]$ = imaginary part of the complex value $x'(n)$.

3. Synthesis Window Function for IMDCT

Proper operation of the IMDCT includes applying a properly designed synthesis window function to the time-domain samples generated by the transform. The time-domain signal samples that are obtained from this windowing operation may be expressed as:

$$\left.\begin{aligned} y'(2n) &= -\mathrm{Im}\left[y\left(\frac{N}{4}+n\right)\right]\cdot h(n) \\ y'(2n+1) &= \mathrm{Re}\left[y\left(\frac{N}{4}-n-1\right)\right]\cdot h\left(\frac{3N}{4}+n\right) \\ y'\left(\frac{N}{2}+2n\right) &= -\mathrm{Re}\left[y(n)\right]\cdot h\left(\frac{N}{4}+n\right) \\ y'\left(\frac{N}{2}+2n+1\right) &= \mathrm{Im}\left[y\left(\frac{N}{2}-n-1\right)\right]\cdot h\left(\frac{N}{2}+n\right) \end{aligned}\right] \quad \text{for } 0 \le n < \frac{N}{4} \tag{6}$$

where $h(n)$ = point n in the synthesis window function; and

$y'(n)$=windowed intermediate sample n.

The windowed intermediate samples y' obtained from expression 6 are the intermediate time-domain samples that could have been obtained by application of the direct IMDCT to a block of the frequency-domain transform coefficients X followed by an application of the synthesis window function h. As explained in the Princen paper cited above, the output time-domain signal samples are obtained by overlapping and adding the windowed interim samples derived from a "current" block of transform coefficients with a set of "previous" windowed interim samples derived from a previous block of transform coefficients. This overlap-add process may be expressed as:

$$x(n) = y'(n) + y'_{prev}(n) \tag{7}$$

where $y'_{prev}(n)$= previous windowed interim samples.

4. Transform-Decomposition Technique

A transform decomposition technique may be used to derive a more efficient method for performing the folded transform for limited bandwidth signals in which some of the transform coefficients in a block of frequency-domain transform coefficients are known to be equal to zero. This decomposition technique consists of expressing the folded transform as an equivalent two-dimensional transform and decomposing this two-dimensional transform into a single-dimension vertical transform followed by a bank of single-dimension horizontal Inverse Discrete Fourier Transforms (IDFT). The vertical transform has a length equal to Q and the bank of horizontal complex IDFT comprises Q transforms each having a length equal to P, where P and Q are integers and the product of P and Q equals the length of the folded transform.

Referring to the preceding discussion of the folding technique, it may be seen that the length of the folded transform is $J = ¼ \cdot N = ½ K$; therefore, $P \cdot Q = J$. The values for P, Q and J are constrained to be powers of two.

The horizontal IDFT and the vertical transform are shown in expressions 8 and 9, respectively:

$$x'(n+Q\cdot m)=\sum_{p=0}^{P-1}U(n,p)\cdot(W_P)^{p\cdot m} \quad \text{for } 0\le n<Q \text{ and } 0\le m<P \tag{8}$$

where $U(n,p)=\sum_{q=0}^{Q-1}X'(P\cdot q+p)\cdot(W_{N/4})^{(P\cdot q+p)\cdot n}$ for $0\le n<Q$. (9)

The transform kernel $W_{N/4}$ in the vertical transform may be calculated using Euler's law:

$$(W_{N/4})^{n\cdot m}=\cos\left(\frac{8\pi\cdot n\cdot m}{N}\right)+j\cdot\sin\left(\frac{8\pi\cdot n\cdot m}{N}\right) \tag{10}$$

Because the direct transform coefficients $X(k)$ represent an audio signal in an LFE channel with a limited bandwidth, only L of these coefficients can have a value other than zero, where L is much less than K. As a result, no more than $\left\lceil\frac{L}{2}\right\rceil$ of the complex-valued frequency-domain transform coefficients $X'(k)$ obtained from the pre-processor stage 401 may have values other than zero and the length of the vertical transform may be reduced. A value M is chosen such that it is the smallest power-of-two equal to or greater than this number $\left\lceil\frac{L}{2}\right\rceil$ and the folding process is modified to derive M complex-valued frequency-domain transform coefficients $X'(k)$ that include the L real-valued direct transform coefficients that can have non-zero values. These M complex-valued frequency-domain transform coefficients are to be processed by the transform stage 402. The size R of the vertical transform is chosen such that $R=\frac{M}{P}$. The transform coefficients $X'(P\cdot r+p)$ are zero for $Pr+p\ge 2R$ or, alternatively, $r\ge R$. By taking these considerations into account, expression 9 can be written as:

$$U(n,p) = \sum_{r=0}^{R-1} \left[X'(P\cdot r + p)\cdot \left(W_{N/4}\right)^{(P\cdot r+p)n} + X'\left(\frac{N}{4} - M + (P\cdot r + p)\right)\cdot \left(W_{N/4}\right)^{\left(\frac{N}{4}-M+(P\cdot r+p)\right)n} \right]$$

$$= \sum_{r=0}^{R-1} X'(P\cdot r + p)\cdot \left(W_{N/4}\right)^{(P\cdot r+p)\cdot n} + \sum_{r=0}^{R-1} X'\left(\frac{N}{4} - M + (P\cdot r + p)\right)\cdot \left(W_{N/4}\right)^{\left(\frac{N}{4}-M+(P\cdot r+p)\right)\cdot n} \qquad (11)$$

for $0 \le n < Q,\ 0 \le p < P.$

5. Integrated Pre-Processor and Vertical Transform

The efficiency of the folding technique combined with the transform-decomposition technique as described above can be improved further by integrating the pre-processor stage 401 and the vertical transform as shown in expression 9 into one process. This is illustrated schematically in Fig. 3.

The length R of the vertical transform may be chosen to be equal to the value M or be a power-of-two sub-multiple $\frac{1}{P}$ of the value M. In an embodiment that conforms to the AC-3 standard mentioned above, the number $½\cdot N$ of real-valued frequency-domain transform coefficients is equal to 256 and the spectral content of the audio signal in the LFE channel can be represented by seven real-valued transform coefficients $X(k)$, where $0 \le k < 7$. The pre-processor stage 401 folds these seven real-valued transform coefficients into four complex-valued transform coefficients that are subsequently processed by the folded transform whose length is $J = ¼\cdot N = 128$. As a result, given four complex-valued transform coefficients in this embodiment, M is equal to four and R can be set equal to either 4, 2 or 1 by setting P equal to 1, 2 or 4, respectively. Because $P\cdot Q = J$, the horizontal transform length Q is equal to 128, 64 and 32 when P is equal to 1, 2 and 4, respectively. Little or no gain in efficiency is achieved when P equals one.

When P is set equal to two, the values obtained from the output of the vertical transform indices do not need to be bit-reversed given the small number of coefficients computed in each of the horizontal transforms. The need for bit reversal of transform indices for the Cooley-Tukey FFT algorithm is well known. Bit reversal is not needed when P is set equal to two, however, because the bit-reversal for a complex DFT of length two produces the same coefficient indexing that is achieved by performing no bit-reversal. This computational advantage is offset by having a larger number of horizontal

transforms to perform. The values for P and Q may be selected in response to various design considerations such as processing limitations in the hardware chosen to implement the processes.

An integration of the process shown in expression 3 with the vertical transform shown in expression 9 can be derived by making substitutions for $X'(k)$ and $\left(W_{N/4}\right)^{(P\cdot r+p)\cdot n}$ in expression 9 according to the expressions 3 and 10, respectively. These substitutions yield the following kernel function for the vertical transform:

$$\left[X\left(\frac{N}{2}-2(P\cdot q+p)-1\right)+j\cdot X\left(2(P\cdot q+p)\right)\right]\cdot \left[-\cos\left(\frac{2\pi\left(8(P\cdot q+p)+1\right)}{8N}\right)-j\cdot\sin\left(\frac{2\pi\left(8(P\cdot q+p)+1\right)}{8N}\right)\right]\cdot \left[\cos\left(\frac{8\pi(P\cdot q+p)\cdot n}{N}\right)+j\cdot\sin\left(\frac{8\pi(P\cdot q+p)\cdot n}{N}\right)\right] \quad (12)$$

for $0\le(P\cdot q+p)<\left(\frac{N}{4}-1\right)$.

The cross product of sine and cosine terms in expression 12 can be rewritten as:

$$\left[-\cos\left(\frac{2\pi(8s+1)}{8N}\right)\cdot\cos\left(\frac{8\pi sn}{N}\right)+\sin\left(\frac{2\pi(8s+1)}{8N}\right)\cdot\sin\left(\frac{8\pi sn}{N}\right)\right]- j\cdot\left[\cos\left(\frac{2\pi(8s+1)}{8N}\right)\cdot\sin\left(\frac{8\pi sn}{N}\right)+\sin\left(\frac{2\pi(8s+1)}{8N}\right)\cdot\cos\left(\frac{8\pi sn}{N}\right)\right]$$
$$=(-\cos\alpha\cdot\cos\beta+\sin\alpha\cdot\sin\beta)-j\cdot(\cos\alpha\cdot\sin\beta+\sin\alpha\cdot\cos\beta)$$
$$=-\cos(\alpha+\beta)-j\cdot\sin(\alpha+\beta) \quad (13)$$

where $s=P\cdot q+p$;

$\alpha=\frac{2\pi(8s+1)}{8N}$; and

$\beta=\frac{8\pi sn}{N}$.

It may be seen that

$$\alpha+\beta=\frac{\pi\left(8s\cdot(1+4n)+1\right)}{4N} \quad (14)$$

which we denote as $I(s,n)$ to simplify the following expressions. Using this notation, expression 11 may be rewritten as:

$$U(n,p)=\sum_{r=0}^{R-1}\left[X\left(\frac{N}{2}-2v-1\right)+j\cdot X(2v)\right]\cdot\left(-\cos\left(I(v,n)\right)-j\cdot\sin\left(I(v,n)\right)\right)+$$
$$\sum_{r=0}^{R-1}\left[X\left(\frac{N}{2}-2u-1\right)+j\cdot X(2u)\right]\cdot\left(-\cos\left(I(u,n)\right)-j\cdot\sin\left(I(u,n)\right)\right) \quad (15)$$

where $v=P\cdot r+p$; and

$$u=\frac{N}{4}-M+(P\cdot r+p).$$

Performing the complex multiply we get:

$$U(n,p)=\sum_{r=0}^{R-1}\left[-X\left(\frac{N}{2}-2v-1\right)\cdot\cos\left(I(v,n)\right)-\right.$$
$$j\cdot X\left(\frac{N}{2}-2v-1\right)\cdot\sin\left(I(v,n)\right)-$$
$$\left.j\cdot X(2v)\cdot\cos\left(I(v,n)\right)+X(2v)\cdot\sin\left(I(v,n)\right)\right]+$$
$$\sum_{r=0}^{R-1}\left[-X\left(\frac{N}{2}-2u-1\right)\cdot\cos\left(I(u,n)\right)-\right.$$
$$j\cdot X\left(\frac{N}{2}-2u-1\right)\cdot\sin\left(I(u,n)\right)-$$
$$\left.j\cdot X(2u)\cdot\cos\left(I(u,n)\right)+X(2u)\cdot\sin\left(I(u,n)\right)\right] \quad (16)$$

The computational complexity of the function $U(n,p)$ can be reduced further by taking advantage of the fact that the frequency-domain coefficients $X(v)$ can be non-zero only for $0\le v<2R$. This reduction is reflected in the following expression that also divides the function into real and imaginary component functions $U_R(n,p)$ and $U_I(n,p)$, respectively, where $U(n,p)=U_R(n,p)+j\cdot U_I(n,p)$:

$$U_R(n,p)=\sum_{r=0}^{R-1}\left[X(2v)\cdot\sin\left(I(v,n)\right)-X(2M-2v-1)\cdot\cos\left(I(u,n)\right)\right]$$
$$U_I(n,p)=\sum_{r=0}^{R-1}\left[-X(2v)\cdot\cos\left(I(v,n)\right)-X(2M-2v-1)\cdot\sin\left(I(u,n)\right)\right] \quad (17)$$

This integration of the pre-processor stage 401 and the vertical transform is illustrated schematically in Fig. 4.

The computational resources required to implement the function $U(n,p)$ or its component functions $U_R(n,p)$ and $U_I(n,p)$ can be reduced by pre-calculating the functions $\sin(I(v,n))$, $\cos(I(v,n))$, $\sin(I(u,n)$ and $\cos(I(u,n)$ for all values of v, u and n. Storing the calculated results in lookup tables requires $4\cdot P\cdot R\cdot Q$ entries, where the factor of four accounts for all combinations of sine, cosine, v and u in expression 17.

The size of the table may be reduced further by 12.5% by recognizing that $I(0,n) = \frac{\pi}{4N}$ for all n. As a result, the number of entries required for all factors of X in expression 17 is on the order of $3.5 \cdot P \cdot R \cdot Q$.

If the size of these tables is larger than desired, their size can be reduced by taking advantage of the fact that many of the entries for $I(v,n)$ in the table have duplicate values due to the periodicity of the sine and cosine basis functions. This reduction in size can be achieved in exchange for additional processing resources needed to lookup entries in table because a more elaborate indexing scheme will be needed to access data in the table.

Other techniques may be used to reduce table-size requirements. For example, if sine and cosine tables already exist in a particular implementation, then only $I(v,n)$ and $I(u,n)$ are needed, which reduces the number of table entries by a factor of two.

C. Implementation

Devices that incorporate various aspects of the present invention may be implemented in a variety of ways including software for execution by a computer or some other device that includes more specialized components such as digital signal processor (DSP) circuitry coupled to components similar to those found in a general-purpose computer. Fig. 5 is a schematic block diagram of a device 70 that may be used to implement aspects of the present invention. The processor 72 provides computing resources. RAM 73 is system random access memory (RAM) used by the processor 72 for processing. ROM 74 represents some form of persistent storage such as read only memory (ROM) for storing programs needed to operate the device 70 and possibly for carrying out various aspects of the present invention. I/O control 75 represents interface circuitry to receive and transmit signals by way of the communication channels 76, 77. In the embodiment shown, all major system components connect to the bus 71, which may represent more than one physical or logical bus; however, a bus architecture is not required to implement the present invention.

In embodiments implemented by a general-purpose computer system, additional components may be included for interfacing to devices such as a keyboard or mouse and a display, and for controlling a storage device 78 having a storage medium such as magnetic tape or disk, or an optical medium. The storage medium may be used to record programs of instructions for operating systems, utilities and applications, and may include programs that implement various aspects of the present invention.

The functions required to practice various aspects of the present invention can be performed by components that are implemented in a wide variety of ways including discrete

logic components, integrated circuits, one or more ASICs and/or program-controlled processors. The manner in which these components are implemented is not important to the present invention.

Software implementations of the present invention may be conveyed by a variety of machine readable media such as baseband or modulated communication paths throughout the spectrum including from supersonic to ultraviolet frequencies, or storage media that convey information using essentially any recording technology including magnetic tape, cards or disk, optical cards or disc, and detectable markings on media including paper.

CLAIMS

1. A method for processing a digital audio signal, wherein the method comprises:

receiving a block of real-valued transform coefficients, wherein the block has a quantity K of real-valued transform coefficients of which only a number L of the real-valued transform coefficients representing spectral components of a limited-bandwidth audio signal, ½ $L < M < K$, and M is a power of two;

applying a first-transform of length R to a block of complex-valued coefficients derived from M complex-valued transform coefficients that include the L real-valued transform coefficients representing spectral components of the limited-bandwidth audio signal, wherein $R = \frac{M}{P}$ and P is a power of two;

applying a bank of Q second transforms of length P to outputs of the first transform; and

deriving a sequence of N real-valued signal samples from outputs of the bank of second transforms, wherein $N = 2 \cdot K$ and the real-valued signal samples represent temporal components of the limited-bandwidth audio signal.

2. The method of claim 1, wherein:

each of the second transforms is equivalent to performing calculations expressed as $x'(n+Q\cdot m) = \sum_{p=0}^{P-1} U(n,p)\cdot (W_P)^{p\cdot m}$ for $0 \le n < Q$ and $0 \le m < P$;

the sequence of real-valued signal samples is derived from the outputs of the bank of second transforms by performing calculations equivalent to

$$y(n) = \left(\mathrm{Re}\left[x'(n)\right] + j\cdot \mathrm{Im}\left[x'(n)\right]\right)\cdot\left(-\cos\left(\frac{2\pi(8n+1)}{8N}\right) - j\cdot\sin\left(\frac{2\pi(8n+1)}{8N}\right)\right);$$

where x' represents the outputs of the second transform;

$U(n,p)$ = a kernel function of the first transform;

$W_P = e^{j\left(\frac{2\pi}{P}\right)}$;

$y(n)$ represents the intermediate signal samples;

$\mathrm{Re}\left[x'(n)\right]$ = real part of $x'(n)$;

$\mathrm{Im}\left[x'(n)\right]$ = imaginary part of $x'(n)$;

j = imaginary operator equal to $\sqrt{-1}$; and

m, n and p are indices used in calculations.

3. The method of claim 2, wherein the first transform is equivalent to performing calculations expressed as

$$U(n,p)=\sum_{r=0}^{R-1}\left[X'(P\cdot r+p)\cdot\left(W_{N/4}\right)^{(P\cdot r+p)n}+X'\left(\frac{N}{4}-M+(P\cdot r+p)\right)\cdot\left(W_{N/4}\right)^{\left(\frac{N}{4}-M+(P\cdot r+p)\right)n}\right]$$

$$=\sum_{r=0}^{R-1}X'(P\cdot r+p)\cdot\left(W_{N/4}\right)^{(P\cdot r+p)\cdot n}+\sum_{r=0}^{R-1}X'\left(\frac{N}{4}-M+(P\cdot r+p)\right)\cdot\left(W_{N/4}\right)^{\left(\frac{N}{4}-M+(P\cdot r+p)\right)\cdot n}$$

for $0\le n<Q$ and $0\le p<P$;

where X represents the real-valued transform coefficients;

$R=\frac{M}{P}$; and

r is an index used in calculations.

4. The method of claim 2, wherein the first transform is equivalent to performing calculations expressed as

$$U(n,p)=\sum_{r=0}^{R-1}\left[-X\left(\frac{N}{2}-2v-1\right)\cdot\cos\left(I(v,n)\right)-\right.$$
$$j\cdot X\left(\frac{N}{2}-2v-1\right)\cdot\sin\left(I(v,n)\right)-$$
$$\left.j\cdot X\left(2v\right)\cdot\cos\left(I(v,n)\right)+X\left(2v\right)\cdot\sin\left(I(v,n)\right)\right]+$$
$$\sum_{r=0}^{R-1}\left[-X\left(\frac{N}{2}-2u-1\right)\cdot\cos\left(I\left(u,n\right)\right)-\right.$$
$$j\cdot X\left(\frac{N}{2}-2u-1\right)\cdot\sin\left(I\left(u,n\right)\right)-$$
$$\left.j\cdot X\left(2u\right)\cdot\cos\left(I\left(u,n\right)\right)+X\left(2u\right)\cdot\sin\left(I\left(u,n\right)\right)\right]$$

for $0\le n<Q$ and $0\le p<P$;

where X represents the real-valued transform coefficients;

$$R=\frac{M}{P};$$

$$I\left(s,n\right)=\frac{\pi\left(8s\cdot\left(1+4n\right)+1\right)}{4N};$$

$$v=P\cdot r+p;$$

$$u=\frac{N}{4}-M+(P\cdot r+p);\text{ and}$$

r is an index used in calculations.

5. The method of claim 2, wherein the first transform is equivalent to performing calculations expressed as

$$U_R\left(n,p\right)=\sum_{r=0}^{R-1}\left[X\left(2v\right)\cdot\sin\left(I(v,n)\right)-X\left(2M-2v-1\right)\cdot\cos\left(I\left(u,n\right)\right)\right]\quad\text{and}$$

$$U_I\left(n,p\right)=\sum_{r=0}^{R-1}\left[-X\left(2v\right)\cdot\cos\left(I(v,n)\right)-X\left(2M-2v-1\right)\cdot\sin\left(I\left(u,n\right)\right)\right]$$

for $0\le n<Q$ and $0\le p<P$;

where X represents the real-valued transform coefficients;

$$R=\frac{M}{P};$$

$$I\left(s,n\right)=\frac{\pi\left(8s\cdot\left(1+4n\right)+1\right)}{4N};$$

$$v=P\cdot r+p;$$

$u = \frac{N}{4} - M + (P \cdot r + p)$; and

r is an index used in calculations.

6. An apparatus for processing a digital audio signal, wherein the apparatus comprises means for performing all steps of the method of any one claims 1 through 5.

7. A storage medium recording a program of instructions that is executable by a device to perform a method for processing a digital audio signal, wherein the method comprises all steps of the method of any one claims 1 through 5.