



Method for obtaining a multi-channel decoder matrix

The present invention relates to a method for obtaining a multi-channel decoder matrix and to the use of such method in decoder devices.

Background

There are basic shortcomings of conventional two-channel sound systems so as to allow enhanced performance in stereophonic presentation and/or multi-lingual capability. It is recognized that the improved presentation performance is desirable not only for applications with accompanying picture but also in case of audio-only applications. A universal and compatible multi-channel sound system applicable to sound and television broadcasting, like satellite or terrestrial TV broadcasting, digital audio broadcasting (terrestrial and satellite), as well as other non-broadcasting media, e.g.,

CATV Cable TV Distribution

CDAD Cable Digital Audio Distribution

ENG Electronic News Gathering (including Satellite News Gathering)

IPC Interpersonal Communications (videoconferencing, video-phone, etc.)

ISM Interactive Storage Media (optical disks, etc.)

NDB Network Database Services (via ATM, etc.)

DSM Digital Storage Media (digital VTR, etc.)

EC Electronic Cinema

HTT Home Television Theatre

ISDN Integrated Service Digital Network

seems to be very attractive to the manufacturer, producer, and consumer.

The ISO/MPEG-Audio Multichannel system is an audio subband coding system, which can be used to transfer high quality digital multi-channel and/or multi-lingual audio information on channels with limited capacity. One of the basic features is the backwards compatibility to ISO 11172-3-IS coded mono, stereo or dual

channel audio programs. It is designed for use in different applications.

As regards stereophonic presentation, the use of an additional center loudspeaker channel C and two surround loudspeaker channels Ls (or LS) and Rs (or RS) is recommended, augmenting the front left and right loudspeaker channels L and R. This reference sound format is referred to as "3/2-stereo" (3 front / 2 surround loudspeaker channels), and requires the transmission of five appropriately formatted sound signals.

For picture accompanying sound applications (e.g. HDTV), the three front loudspeaker channels ensure sufficient directional stability and clarity of the picture related frontal images, according to the common practice in the cinema. The dominant benefit is the "stable center", which is guaranteed at any location of the listener and important for most of the dialogue.

Additionally for audio-only applications the 3/2-stereo format has been found to be the appropriate improvement of two-channel stereophony. The addition of one pair of surround loudspeaker channels allows improved realism of auditory ambience.

In basic terms, the transmission of the five audio signals of a 3/2 sound system requires five transmission channels (although, in the context of bit-rate reduced signals, these are not necessarily totally independent). In order that two of the transmitted signals can provide a stereo service on their own, the source sound signals are generally combined in a linear matrix prior to encoding. These combined signals (and their transmission channels) are identified by the notation T1, T2, T3, T4 and T5. For compatibility reasons, an ISO 11172-3 IS audio decoder properly decodes the basic stereo information, consisting of basic left and right channels. The signals in these channels constitute an appropriate downmix of the audio information in all channels. The appropriate downmix equations are given by the equations (T1 and T2 correspond to Lo and Ro):

$$Lo = L + x * C + y * Ls \quad (1)$$

$$Ro = R + x * C + y * Rs \quad (2)$$

The parameter *tc_allocation* contains information on the transmission channel allocation in a subband group:



tc_allocation	T3	T4	T5
0	L	R	C
1	L	R	LS
2	L	R	RS
3	L	C	LS
4	L	C	RS
5	L	LS	RS
6	R	C	LS
7	R	C	RS
8	R	LS	RS
9	C	LS	RS
10	L	R	-
11	LS	RS	-
12	L	-	-
13	R	-	-
14	C	-	-
15	-	-	-

Further details are described in "ISO 11172-3 compatible low bit rate multi-channel audio coding and conventional stereo coding at lower sampling frequencies", ISO/IEC JTC1/SC29/WG11 N0403, MPEG93/479, 11. June 1993.

Invention

It is an object of the invention to obtain a decoder matrix in a decoder device without storing a decoder matrix table containing multiple of these matrices. This object is reached by the method disclosed in claim 1.

The values required for equations (1) and (2) can be represented by a multichannel (MC) encoder matrix. The following matrix may serve as an example:

This Sheet Has Been Replaced





$$\begin{pmatrix} Lo \\ Ro \\ T3 \\ T4 \\ T5 \end{pmatrix} = \begin{pmatrix} 1 & 0 & x & y & 0 \\ 0 & 1 & x & 0 & y \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} L \\ R \\ C \\ Ls \\ Rs \end{pmatrix}$$

In a decoder, the inverse matrix is required to extract the values L, R, C, Ls and Rs. The variation of these parameters would allow a huge amount of different decoder matrices even if a lot of them are not useful or not possible.

Some (e.g. 10) bits are transmitted, which represent e.g. the values x and y, center YES/NO, surround NO/MONO/STEREO, content of T3, T4 and T5. These bits can be used to address in the decoder a large memory in which the corresponding inverse matrices are listed.

It is too expensive to implement all decoder matrix combinations in a memory table within a consumer decoder device, therefore the number of combinations must be reduced.

Advantageously, instead of addressing a decoder matrix stored in a decoder memory, at first the current encoder matrix can be reconstructed in the decoder using the related transmitted information. Then, the required decoder matrix can be computed from the reconstructed encoder matrix. This is possible due to the fact that the necessary computation time is significantly lower than the required computation time for a decoder audio signal filter.

In principle, the inventive method is suited for obtaining a multi-channel decoder matrix in a system having indication information concerning the kind of the signal to be decoded, wherein said signal has previously been coded using a corresponding encoder matrix and said indication information comprises channel on/off data, channel factor data, and channel allocation data, and comprising the steps:

- reconstructing from said indication information the size of said encoder matrix and inserting the corresponding 0/1 elements into said encoder matrix;
- inserting the elements related to said channel factor data into said encoder matrix;

- evaluating said channel allocation data and inserting it into the remaining positions of said encoder matrix;
- performing a matrix inversion of the reconstructed encoder matrix to get said decoder matrix.

Drawings

Preferred embodiments of the invention are described with reference to the accompanying drawings, which show in:

Fig. 1 arrangement of main, center and surround channels;

Fig. 2 apparatus for computing a decoder matrix.

Preferred embodiments

In Fig. 1 the left and right main channels L and R are depicted together with an additional center channel C and surround channels Ls and Rs. For compatibility reasons (MPEG1/MPEG2), not these signals but the signals Lo, Ro (which include the significant audio information) and T3, T4 and T5 are transmitted.

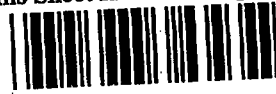
The decoder matrices for each of the 11 subband groups per frame are indicated by the following header bits:

Bitstream	Information	No./frame	Bits	Total
MPEG1	MODE	1	2	2
MPEG2	Center	1	1	1
- "-	Surround	1	2	2
- "-	Presentation Matrix	1	2	2
- "-	tc_allocation *)	[1 or] 11	3	33
Sum			10	40

*) without dynamic range control

This Sheet Has Been Replaced





This number of bits leads theoretically to up to 1024 different decoder matrices. To reduce this large number, e.g. the multichannel feature can be restricted to the MPEG1 stereo and *joint_stereo* mode. But this means that combinations like 1/2 and 1/1 or 1/0 + 1/0 + 2/0 are not allowed. The remaining combinations can be seen from the following list:

Signal Configuration table:

MPEG1 Mode	MPEG2 Center	Surround	Signal Configuration
Stereo &	yes	no	LRC (3/0)
		mono	LRCS (3/1)
		stereo	LRCLsRs (3/2)
		second stereo	LRCL ₂ R ₂ (3/0 + 2/0)

Joint Stereo	no	no	LR (no MC)
		mono	LRS (2/1)
		stereo	LRLsRs (2/2)
		second stereo	LRL ₂ R ₂ (2/0 + 2/0)

Dual	yes	no	M ₁ M ₂ C
		mono	M ₁ M ₂ CS
		stereo	M ₁ M ₂ CLsRS
		second stereo	M ₁ M ₂ CL ₂ R ₂

	no	no	M ₁ M ₂ (no MC)
		mono	M ₁ M ₂ S
		stereo	M ₁ M ₂ LSRS
		second stereo	M ₁ M ₂ L ₂ R ₂

Single	yes	no	MC
		mono	MCS
		stereo	MCLsRS
		second stereo	MCL ₂ R ₂

	no	no	M (no MC)
		mono	MC



stereo _____ MLRS _____
 second stereo _____ ML2R2 _____

Remark: 'no MC' does not mean 'no ML'

From the MPEG bitstream the presentationmatrix information can be used to find out, how the basic stereo information Lo and Ro has been generated in the encoder (*single_channel* and *dual_channel* are not allowed). Again, all combinations are listed in the next table. At present, one value of the presentation matrix is not fixed yet, but is set to -6 dB in the next table.

Compatible Basic Stereo table:

Signal Config.	Presentation Matrix	Xo
LRCLsRs (3/2)	-3 dB S.; -3 dB C.	$X + 0.707 C + 0.707 Xs$
	-6 dB S.; -3 dB C.	$X + 0.707 C + 0.5 Xs$
	$-\infty$ dB S.; -3 dB C.	$X + 0.707 C$
	$-\infty$ dB S.; $-\infty$ dB C.	X

LRCS (3/1)	-3 dB S.; -3 dB C.	$X + 0.707 C + 0.707 S$
	-6 dB S.; -3 dB C.	$X + 0.707 C + 0.5 S$
	$-\infty$ dB S.; -3 dB C.	$X + 0.707 C$
	$-\infty$ dB S.; $-\infty$ dB C.	X

LRCL ₂ R ₂ (3/0+2/0) & LRC (3/0)	-3 dB S.; -3 dB C.	$X + 0.707 C$
	-6 dB S.; -3 dB C.	$X + 0.707 C$
	$-\infty$ dB S.; -3 dB C.	$X + 0.707 C$
	$-\infty$ dB S.; $-\infty$ dB C.	X

LRLsRs (2/2)	-3 dB S.; -3 dB C.	$X + 0.707 Xs$
	-6 dB S.; -3 dB C.	$X + 0.5 Xs$
	$-\infty$ dB S.; -3 dB C.	X
	$-\infty$ dB S.; $-\infty$ dB C.	X

LRS	-3 dB S.; -3 dB C.	X + 0.707 S
(2/1)	-6 dB S.; -3 dB C.	X + 0.5 S
	$-\infty$ dB S.; -3 dB C.	X
	$-\infty$ dB S.; $-\infty$ dB C.	X

LRL ₂ R ₂	-3 dB S.; -3 dB C.	X
(2/0 + 2/0)	-6 dB S.; -3 dB C.	X
	$-\infty$ dB S.; -3 dB C.	X
	$-\infty$ dB S.; $-\infty$ dB C.	X

Now, all information is collected which is useful for a hole frame. Finally, the information about the allocation of the transport channels is needed, i.e. which audio channel is in which transport channel for a subband group. This information is indicated by the *tc_allocation*. For each signal configuration an other list is used. The table below (originally elaborated by PHILIPS) shows that again not all possibilities are allowed.

tc_allocation table:

Signal Config.	Presentation Matrix	Tc_ all.	(T1)	(T2)	T3	T4	T5	MTX No
LRCLsRs (3/2)	-3dB S; -3dB C	0	L	R	C	LS	RS	1/1
		1	C	R	L	LS	RS	2/2
		2	L	C	R	LS	RS	3/3
		3	Ls	R	C	L	RS	4/4
		4	L	Rs	C	LS	R	5/5
		5	Ls	Rs	C	L	R	6/6
		6	Ls	C	R	L	RS	7/7
		7	C	Rs	L	LS	R	8/8
		8	-	-				(x/x)
		9	-	-				(x/x)
		10	-	-				(x/x)
		11	-	-	dyn.crosstalk			(x/x)
		12	-	-				(x/x)
		13	-	-				(x/x)
		14	-	-				(x/x)

This Sheet Has Been Replaced



15 - - (x/x)

-6dB S; -3dB C	0	L	R	C	LS	RS	1/9
	1	C	R	L	LS	RS	2/10
	2	L	C	R	LS	RS	3/11
	3	Ls	R	C	L	RS	4/12
	4	L	Rs	C	LS	R	5/13
	5	Ls	Rs	C	L	R	6/14
	6	Ls	C	R	L	RS	7/15
	7	C	Rs	L	LS	R	8/16
	x	-	-	dyn.crosstalk			(x/x)

-∞dB S; -3dB C	0	L	R	C	LS	RS	1/17
	1	C	R	L	LS	RS	2/18
	2	L	C	R	LS	RS	3/19
	3	~~Ls~~	~~R~~	~~C~~	~~L~~	~~RS~~	~~4~~
	4	~~L~~	~~Rs~~	~~C~~	~~LS~~	~~R~~	~~5~~
	5	~~Ls~~	~~Rs~~	~~C~~	~~L~~	~~R~~	~~6~~
	6	~~Ls~~	~~C~~	~~R~~	~~L~~	~~RS~~	~~7~~
	7	~~C~~	~~Rs~~	~~L~~	~~LS~~	~~R~~	~~8~~
	x	-	-	dyn. cross talk			(x/x)

-∞dB S; -∞dB C	0	L	R	C	LS	RS	1/20
	1	~~C~~	~~R~~	~~L~~	~~LS~~	~~RS~~	~~2~~
	2	~~L~~	~~C~~	~~R~~	~~LS~~	~~RS~~	~~3~~
	3	~~Ls~~	~~R~~	~~C~~	~~L~~	~~RS~~	~~4~~
	4	~~L~~	~~Rs~~	~~C~~	~~LS~~	~~R~~	~~5~~
	5	~~Ls~~	~~Rs~~	~~C~~	~~L~~	~~R~~	~~6~~
	6	~~Ls~~	~~C~~	~~R~~	~~L~~	~~RS~~	~~7~~
	7	~~C~~	~~Rs~~	~~L~~	~~LS~~	~~R~~	~~8~~
	x	-	-	dyn. cross talk			(x/x)

LRCS	-3dB S; -3dB C	0	L	R	C	S	-	9/21
(3/1)		1	C	R	L	S	-	10/22
		2	L	C	R	S	-	11/23
		3	S	R	C	L	-	12/24
		4	L	S	C	R	-	13/25
		x	-	-	dyn. cross talk			(x/x)

This Sheet Has Been Replaced





-6dB S; -3dB C	0	L	R	C	S	-	9/26		
	1	C	R	L	S	-	10/27		
	2	L	C	R	S	-	11/28		
	3	S	R	C	L	-	12/29		
	4	L	S	C	R	-	13/30		
	x	-	-	dyn. cross talk				(x/x)	

-∞dB S; -3dB C	0	L	R	C	S	-	9/31		
	1	C	R	L	S	-	10/32		
	2	L	C	R	S	-	11/33		
	3	~~S~~	~~R~~	~~C~~	~~L~~		~~12~~		
	4	~~L~~	~~S~~	~~C~~	~~R~~		~~13~~		
	x	-	-	dyn. cross talk				(x/x)	

-∞dB S; -∞dB C	0	L	R	C	S	-	9/34		
	1	~~C~~	~~R~~	~~L~~	~~S~~		~~10~~		
	2	~~L~~	~~C~~	~~R~~	~~S~~		~~11~~		
	3	~~S~~	~~R~~	~~C~~	~~L~~		~~12~~		
	4	~~L~~	~~S~~	~~C~~	~~R~~		~~13~~		
	x	-	-	dyn. cross talk				(x/x)	

LRCL ₂ R ₂ (3/0 + 2/0) & LRC (3/0)	0	L	R	C	(L2)	(R2)	14/35		
	1	C	R	L	(L2)	(R2)	15/36		
	2	L	C	R	(L2)	(R2)	16/37		
	x	-	-	dyn. cross talk				(x/x)	

-6dB S; -3dB C	0	L	R	C	(L2)	(R2)	14/38		
	1	C	R	L	(L2)	(R2)	15/39		
	2	L	C	R	(L2)	(R2)	16/40		
	x	-	-	dyn. cross talk				(x/x)	

-∞dB S; -3dB C	0	L	R	C	(L2)	(R2)	14/41		
	1	C	R	L	(L2)	(R2)	15/42		
	2	L	C	R	(L2)	(R2)	16/43		
	x	-	-	dyn. cross talk				(x/x)	

-∞dB S; -∞dB C	0	L	R	C	(L2)	(R2)	14/44		

11

~~1~~	~~C~~	~~R~~	~~L~~	~~(L2)~~	~~(R2)~~	~~15~~
~~2~~	~~L~~	~~C~~	~~R~~	~~(L2)~~	~~(R2)~~	~~16~~
x	-	-	dyn. cross talk			(x/x)

LRLsRs	-3dB S; -3dB C	0	L	R	LS	RS	-	17/45
(2/2)		3	Ls	R	L	RS	-	18/46
		4	L	Rs	LS	R	-	19/47
		5	Ls	R	L	R	-	20/48
		x	-	-	dyn. cross talk			(x/x)

	-6dB S; -3dB C	0	L	R	LS	RS	-	17/49
		3	Ls	R	L	RS	-	18/50
		4	L	Rs	LS	R	-	19/51
		5	Ls	R	L	R	-	20/52
		x	-	-	dyn. cross talk			(x/x)

	-∞dB S; -3dB C	0	L	R	LS	RS	-	17/53
		~~3~~	~~Ls~~	~~R~~	~~L~~	~~RS~~	~~-~~	~~18~~
		~~4~~	~~L~~	~~Rs~~	~~LS~~	~~R~~	~~-~~	~~19~~
		~~5~~	~~Ls~~	~~R~~	~~L~~	~~R~~	~~-~~	~~20~~
		x	-	-	dyn. cross talk			(x/x)

	-∞dB S; -∞dB C	0	L	R	LS	RS	-	17/54
		~~3~~	~~Ls~~	~~R~~	~~L~~	~~RS~~	~~-~~	~~18~~
		~~4~~	~~L~~	~~Rs~~	~~LS~~	~~R~~	~~-~~	~~19~~
		~~5~~	~~Ls~~	~~R~~	~~L~~	~~R~~	~~-~~	~~20~~
		x	-	-	dyn. cross talk			(x/x)

LRS	-3dB S; -3dB C	0	L	R	S	-	-	21/55
(2/1)		3	S	R	L	-	-	22/56
		4	L	S	R	-	-	23/57
		x	-	-	dyn. cross talk			(x/x)

	-6dB S; -3dB C	0	L	R	S	-	-	21/58
		3	S	R	L	-	-	22/59
		4	L	S	R	-	-	23/60
		x	-	-	dyn. cross talk			(x/x)

This Sheet Has Been Replaced



12

-∞dB S; -∞dB C	0	L	R	S	-	-	21/61
	3	~~S~~	~~R~~	~~L~~			22
	4	~~L~~	~~S~~	~~R~~			23
	x	-	-	dyn. cross talk			(x/x)

-∞dB S; -∞dB C	0	L	R	S	-	-	21/62
	3	~~S~~	~~R~~	~~L~~			22
	4	~~L~~	~~S~~	~~R~~			23
	x	-	-	dyn. cross talk			(x/x)

LR	-3dB S; -3dB C	no	L	R	(L2)	(R2)	- no
(2/0)	-6dB S; -3dB C	no	L	R	(L2)	(R2)	- no
LRL ₂ R ₂	-∞dB S; -3dB C	no	L	R	(L2)	(R2)	- no
(2/0 +2/0)	-∞dB S; -∞dB C	no	L	R	(L2)	(R2)	- no

With the restriction to MPEG 1 *stereo* and *joint_stereo* only, no further matrices seem to be possible. So, the storage of 23 types leading to 62 real decoder matrices without dynamic cross talk are required.

The last row in the list above gives a matrix type/real number. All 23 encoder and decoder matrixes types are listed in the next list.

This Sheet Has Been Replaced



Encoder and decoder matrices:

(the second matrix column is received by matrix inversion)

Example:

$$\begin{bmatrix} L_0 \\ R_0 \\ T_3 \\ T_4 \\ T_5 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x & y & 0 \\ 0 & 1 & x & 0 & y \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} R \\ L \\ C \\ L_s \\ R_s \end{bmatrix}$$

$$x = \frac{1}{\sqrt{2}}$$

$$y_1 = \frac{1}{\sqrt{2}} \quad y_2 = \frac{1}{2}$$

$$1) \begin{bmatrix} 1 & 0 & x & y & 0 \\ 0 & 1 & x & 0 & y \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & -x & -y & 0 \\ 0 & 1 & -x & 0 & -y \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x = 0.70711$$

$$y_1 = 0.70711 \quad y_2 = 0.5$$

$$2) \begin{bmatrix} 1 & 0 & x & y & 0 \\ 0 & 1 & x & 0 & y \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ -1 & 1 & 1 & y & -y \\ 1 & 0 & -1 & -y & 0 \\ \bar{x} & 0 & \bar{x} & \bar{x} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\frac{y_1}{x} = 1 \quad \frac{y_2}{x} = 0.70711$$

$$\frac{1}{\bar{x}} = 1.41421$$

$$3) \begin{bmatrix} 1 & 0 & x & y & 0 \\ 0 & 1 & x & 0 & y \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -1 & 1 & -y & y \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & -y \\ 0 & \bar{x} & \bar{x} & 0 & \bar{x} \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$4) \begin{bmatrix} 1 & 0 & x & y & 0 \\ 0 & 1 & x & 0 & y \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & -x & 0 & -y \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & x & -1 \\ y & 0 & y & y & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\frac{x}{y_1} = 1 \quad \frac{x}{y_2} = 1.41421$$

$$\frac{1}{y_1} = 1.41421 \quad \frac{1}{y_2} = 2$$

This Sheet Has Been Replaced



14

$$5) \begin{bmatrix} 1 & 0 & x & y & 0 \\ 0 & 1 & x & 0 & y \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & -x & -y & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & \frac{1}{y} & -\frac{1}{y}x & 0 & -\frac{1}{y} \end{bmatrix}$$

$$6) \begin{bmatrix} 1 & 0 & x & y & 0 \\ 0 & 1 & x & 0 & y \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ \frac{1}{y} & 0 & -\frac{1}{y}x & -\frac{1}{y} & 0 \\ 0 & \frac{1}{y} & -\frac{1}{y}x & 0 & -\frac{1}{y} \end{bmatrix}$$

$$7) \begin{bmatrix} 1 & 0 & x & y & 0 \\ 0 & 1 & x & 0 & y \\ 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & -1 & 0 & -\frac{y}{x} \\ \frac{1}{y} & -\frac{1}{y} & \frac{1}{y} & -\frac{1}{y} & 1 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$8) \begin{bmatrix} 1 & 0 & x & y & 0 \\ 0 & 1 & x & 0 & y \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ \frac{1}{x} & 0 & -\frac{1}{x} & -\frac{y}{x} & 0 \\ 0 & 0 & 0 & 1 & 0 \\ -\frac{1}{y} & \frac{1}{y} & \frac{1}{y} & 1 & -\frac{1}{y} \end{bmatrix}$$

This Sheet Has Been Replaced



Example:

$$\begin{bmatrix} L_0 \\ R_0 \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 1 & x & 0 & y \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} L \\ R \\ C \\ S \end{bmatrix}$$

$$9) \begin{bmatrix} 1 & 0 & x & y \\ 0 & 1 & x & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & -x & -y \\ 0 & 1 & -x & -y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$10) \begin{bmatrix} 1 & 0 & x & y \\ 0 & 1 & x & y \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 & 0 \\ -1 & 1 & 1 & 0 \\ \frac{1}{x} & 0 & -\frac{1}{x} & -\frac{y}{x} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$11) \begin{bmatrix} 1 & 0 & x & y \\ 0 & 1 & x & y \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{x} & -\frac{1}{x} & -\frac{y}{x} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$12) \begin{bmatrix} 1 & 0 & x & y \\ 0 & 1 & x & y \\ 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 1 \\ -1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ \frac{1}{y} & 0 & -\frac{x}{y} & -\frac{1}{y} \end{bmatrix}$$

$$13) \begin{bmatrix} 1 & 0 & x & y \\ 0 & 1 & x & y \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -1 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{y} & -\frac{x}{y} & -\frac{1}{y} \end{bmatrix}$$

Example:

$$\begin{bmatrix} L_0 \\ R_0 \\ T_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & x \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} L \\ R \\ C \end{bmatrix}$$

$$14) \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & x \\ 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & -x \\ 0 & 1 & -x \\ 0 & 0 & 1 \end{bmatrix}$$

$$15) \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & x \\ 1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 \\ -1 & 1 & 1 \\ \frac{1}{x} & 0 & -\frac{1}{x} \end{bmatrix}$$

$$16) \begin{bmatrix} 1 & 0 & x \\ 0 & 1 & x \\ 0 & 1 & 0 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -1 & 1 \\ 0 & 0 & 1 \\ 0 & \frac{1}{x} & -\frac{1}{x} \end{bmatrix}$$

This Sheet Has Been Replaced



Example:

$$\begin{bmatrix} L_o \\ R_o \\ T_3 \\ T_4 \end{bmatrix} = \begin{bmatrix} 1 & 0 & y & 0 \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} L \\ R \\ L_s \\ R_s \end{bmatrix}$$

$$17) \begin{bmatrix} 1 & 0 & y & 0 \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -y & 0 \\ 0 & 1 & 0 & -y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$18) \begin{bmatrix} 1 & 0 & y & 0 \\ 0 & 1 & 0 & y \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & -y \\ \frac{1}{y} & 0 & -\frac{1}{y} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$19) \begin{bmatrix} 1 & 0 & y & 0 \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -y & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & \frac{1}{y} & 0 & -\frac{1}{y} \end{bmatrix}$$

$$20) \begin{bmatrix} 1 & 0 & y & 0 \\ 0 & 1 & 0 & y \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{1}{y} & 0 & -\frac{1}{y} & 0 \\ 0 & \frac{1}{y} & 0 & -\frac{1}{y} \end{bmatrix}$$

Example:

$$\begin{bmatrix} L_o \\ R_o \\ T_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & y \\ 0 & 1 & y \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} L \\ R \\ S \end{bmatrix}$$

$$21) \begin{bmatrix} 1 & 0 & y \\ 0 & 1 & y \\ 0 & 0 & 1 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -y \\ 0 & 1 & -y \\ 0 & 0 & 1 \end{bmatrix}$$

$$22) \begin{bmatrix} 1 & 0 & y \\ 0 & 1 & y \\ 1 & 0 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 1 \\ -1 & 1 & 1 \\ \frac{1}{y} & 0 & -\frac{1}{y} \end{bmatrix}$$

$$23) \begin{bmatrix} 1 & 0 & y \\ 0 & 1 & y \\ 0 & 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & -1 & 1 \\ 0 & 0 & 1 \\ 0 & \frac{1}{y} & -\frac{1}{y} \end{bmatrix}$$

This Sheet Has Been Replaced



To avoid the storage of these matrices in the decoder, from the header of the MPEG bitstream the elements and the size of the matrix can be calculated. For example:

The channel on/off data are

MPEG1 MODE:	stereo	L+R
MPEG2 Center:	yes	C
MPEG2 Surround:	mono	S

I.e. 4 channels:

T1		x	x	x	x		L
T2	=	x	x	x	x	*	R
T3		x	x	x	x		C
T4		x	x	x	x		S

From the above channel on/off data the matrix elements

X_{11}, X_{12}, X_{21} and X_{22} are fixed:

T1		1	0	x	x		L
T2	=	0	1	x	x	*	R
T3		x	x	x	x		C
T4		x	x	x	x		S

The *presentationmatrix* (channel factor data) gives useful information about the factors of the basic stereo channels. For example:

Surround: -6dB = 0.5

Center: -3dB = 0.707

I.e. the matrix looks as followed:

T1		1	0	0.7	0.5		L
T2	=	0	1	0.7	0.5	*	R
T3		x	x	x	x		C
T4		x	x	x	x		S

Now, the matrix lines 3 and 4 are still missing. One can get this information from the *tc_allocation* table (channel allocation data), e.g. *tc_allocation*=2.

This Sheet Has Been Replaced



Then, the encoder matrix is:

T1		1	0	0.7	0.5	L
T2	=	0	1	0.7	0.5	* R
T3		0	1	0	0	C
T4		0	0	0	1	S

The decoder matrix can be obtained by computing the inverse encoder matrix. Matrix $\sqrt{\quad}$ can be carried out e.g. with Gaussian Elimination Algorithms. The following program gives an example: Inverse

/*

Matrix Inversion with Gauss elimination

see (J.C.Nash, "Compact numerical methods for computers", Page 62ff)

*/

#include <stdlib.h>

#include <stdio.h>

#include <math.h>

#include <float.h>

/*Max Matrix Size */

#define M 5

/* prototype(s) */

void help(void);

void gauss_elimination(int N, float A[] [2*M+1]);

/******

void main(int argc, char *argv[])

{

float A[M+1] [2*M+1];

int i,j,N;

FILE *infile;

printf(" \n");

printf(" %s: Matrix Inversion TCE \n", __FILE__);

if(argc != 2) {

This Sheet Has Been Replaced



```
    help();
    exit(1);
}

printf(" The Datafile is : \"%s\" \n",argv[1]);

if ((infile = fopen(argv[1],"r")) == NULL){
    printf(" (%d) File Open Error ",__LINE__);
    exit(1);
}

fscanf(infile,"%d",&N);
if (N > M) {
    printf(" (%d) Matrix Size to large ",__LINE__);
    exit(1);
}

/* Read Matrix */
for(i=1;i<=N;i++)
    for(j=1;j<=N;j++)
        fscanf(infile,"%f",&A[i][j]);

fclose(infile);

/* Input matix with result matix, here unity matrix */
for(i=1;i<=N;i++)
    for(j=1;j<=N;j++){
        A[i][N+j] = 0;
        if(i==j)A[i][j+N] = 1;}

/* Output */
printf(" Input matrix \n");
for(i=1;i<=N;i++){
    for(j=1;j<=N;j++)
        printf("%10.6f ",A[i][j]);
    printf("\n");
}
```

This Sheet Has Been Replaced

```

/* Calculation */
gauss_elimination(N,A);

/* Output */
printf(" Invers Matrix \n");
for(i=1;i<=N;i++){
    for(j=1;j<=N;j++){
        printf("%10.6f ",A[i][j+N]);
        printf("\n");
    }
}

/*****
void gauss_elimination(int N,float A[][2*M+1])
/* Gauss elimination with partial pivoting */
{
    int n,p;
    int j,k,h,i;
    float D, tol,s;

    n = N;
    p = N; /* Spezial case */
    D = 1.0;
    tol = FLT_EPSILON; /* 1.19e-7 or DBL_EPSILON*/

    for (j=1;j<=n-1;j++){
        /* Find the pivot */
        s = fabs(A[j][j]); k = j;
        for (h=j+1;h<=n;h++){
            if (fabs(A[h][j]) > s){
                s = fabs(A[h][j]);
                k = h;
            }
        }
        /* end loop h */
        if (k != j ){
            for (i = j; i<= n+p;i++){
                s = A[k][i]; A[k][i] = A[j][i]; A[j][i] = s;
            } /* end loop i */
            D = -D;

```



```

    } /* end if(k != j) */

    D = D * A[j][j];
    if (fabs(A[j][j]) <= tol) {
        printf (" \a \n (%d) Matrix is computationally singular
\n", __LINE__);
        exit(100);
    }
    for (k=j+1; k<= n; k++){
        A[k][j] = A[k][j] / A[j][j];
        for (i=j+1; i<=n+p; i++)
            A[k][i] = A[k][i] - A[k][j] * A[j][i];
    } /* end loop k */
} /* end loop j */
D = D * A[n][n];
if (fabs(A[n][n]) <= tol) {
    printf(" \a \n (%d) Matrix is computationally singular
\n", __LINE__);
    exit(101);
}

/* Back-substitution */
for(i=n+1; i<= n+p; i++){
    A[n][i] = A[n][i] / A[n][n];
    for (j=n-1; j>=1; j--){
        s = A[j][i];
        for (k=j+1; k<=n; k++){
            s = s - A[j][k] * A[k][i];
        } /* end loop k */
        A[j][i] = s / A[j][j];
    } /*end loop j */
} /* end loop i */
}

/*****/
void help()
{
    float  A[4][4];
    int i,j;

```



```

int N;
N = M;

printf(" \n please typ \"programname datafile\" \n");
printf(" The datafile must be in the following form: \n");
printf(" \tN                <-- SIZE of Matrix
(2..%d)\n",N);
printf(" \tA11 A12 A13 ..  <- Matrix Elements\n");
printf(" \tA21 A22 A23 ..\n");
printf(" \t..  ..  ..  ..\n");

/* TestData */
printf(" Testdata: \n");
A[1][1] = 1; A[1][2] = 0; A[1][3] = sqrt(0.5);
A[2][1] = 0; A[2][2] = 1; A[2][3] = sqrt(0.5);
A[3][1] = 1; A[3][2] = 0; A[3][3] = 0;
printf(" 3\n");
for(i=1;i<=3;i++){
    for(j=1;j<=3;j++){
        printf("%18.14f ",A[i][j]);
    }
    printf("\n");
}

printf(" \nInverse Matrix: \n");
A[1][1] = 0;          A[1][2] = 0; A[1][3] = 1;
A[2][1] = -1;         A[2][2] = 1; A[2][3] = 1;
A[3][1] = sqrt(2.0);  A[3][2] = 0; A[3][3] = sqrt(2.0);
for(i=1;i<=3;i++){
    for(j=1;j<=3;j++){
        printf("%18.14f ",A[i][j]);
    }
    printf("\n");
}
}

/***** END *****/

```

This routine needs 258 steps (including user messages and inputs) with a debugger to invert an encoder matrix with a size of 5*5 on a PC.



In the DSP32C library exists the function "_matinf" which needs approx. 90 μ s for one 5*5 matrix inversion.

This leads to a total computing time of approx. 1ms for 11 full matrix inversions.

For comparison: The MPEG decoder filter for 5 channels needs approx. 30ms computing time. I.e. the workload of the matrix inversion is only about 3.3% of the filter workload.

Advantageously, the computation of the decoder matrix within the decoder does neither need the storage of 62 matrices nor need any restriction of the quantity of possible encoder matrices.

Fig. 2 shows a block diagram for decoder matrix computations. The computations can be made using data extractor means 21, which extract from the MPEG data stream 20 the header information comprising the required input parameters, and memory means 22 in which the respective computation software operating on these input parameters for constructing the decoder matrix is stored and the output data of which are input together with the transmitted audio data of the MPEG data stream 20 to fast processor means 23 which apply the reconstructed decoder matrix to the audio data fed to a respective loudspeaker arrangement.

The invention can be used in MPEG TV receivers and in the fields listed in the background paragraph.

This Sheet Has Been Replaced



Claims

1. Method for obtaining a multi-channel (L, R, C, Ls, Rs) decoder matrix in a system having indication information concerning the kind of the signal (20) to be decoded, wherein said signal has previously been coded using a corresponding encoder matrix and said indication information comprises channel on/off data, channel factor data, and channel allocation data, characterised in the steps:
 - reconstructing from said indication information the size of said encoder matrix and inserting the corresponding 0/1 elements into said encoder matrix;
 - inserting the elements related to said channel factor data into said encoder matrix;
 - evaluating said channel allocation data and inserting it into the remaining positions of said encoder matrix;
 - performing a matrix inversion of the reconstructed encoder matrix to get said decoder matrix.
2. Usage of the method according to claim 1 in a TV receiver, a VCR, a CD player and/or a computer which supplies the decoded signal to display means.



This Sheet Has Been Replaced

