



(12) **DEMANDE DE BREVET EUROPEEN**

(43) Date de publication:  
**09.09.2020 Bulletin 2020/37**

(51) Int Cl.:  
**G10L 19/008 (2013.01)**

(21) Numéro de dépôt: **19305254.5**

(22) Date de dépôt: **05.03.2019**

(84) Etats contractants désignés:  
**AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR**  
 Etats d'extension désignés:  
**BA ME**  
 Etats de validation désignés:  
**KH MA MD TN**

(72) Inventeurs:  
 • **RAGOT, Stéphane**  
**92326 CHÂTILLON CEDEX (FR)**  
 • **MAHE, Pierre**  
**92326 CHÂTILLON CEDEX (FR)**

(74) Mandataire: **Plasseraud IP**  
**66, rue de la Chaussée d'Antin**  
**75440 Paris Cedex 09 (FR)**

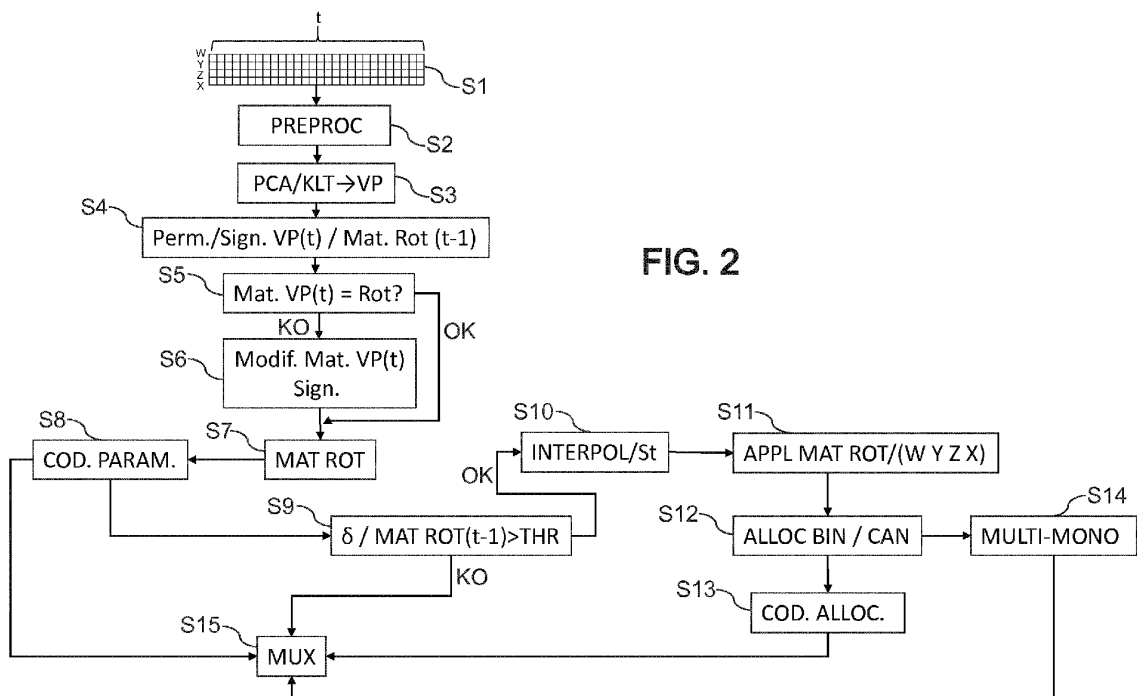
(71) Demandeur: **ORANGE**  
**75015 Paris (FR)**

(54) **CODAGE AUDIO SPATIALISÉ AVEC INTERPOLATION ET QUANTIFICATION DE ROTATIONS**

(57) L'invention concerne le codage en compression de signaux sonores formant une succession dans le temps de trames d'échantillons, dans chacun de N canaux en représentation ambisonique d'ordre supérieur à 0, le procédé comportant :

- former, à partir des canaux pour une trame courante, une matrice de covariance entre canaux et rechercher (S3) des vecteurs propres de la matrice de covariance pour obtenir une matrice de vecteurs propres,

- tester (S5) la matrice de vecteurs propres pour vérifier qu'elle représente une rotation dans un espace de dimension N et corriger (S6) sinon la matrice de vecteurs propres jusqu'à obtenir une matrice de rotation, pour la trame courante, et
- appliquer ladite matrice de rotation (S7) aux signaux des N canaux avant un encodage par canaux séparés desdits signaux.



**FIG. 2**

## Description

**[0001]** La présente invention concerne le codage/décodage de données sonores spatialisées, notamment en contexte ambiophonique (noté ci-après également « ambisonique »).

**[0002]** Les codeurs/décodeurs (ci-après appelés « codecs ») qui sont utilisés actuellement en téléphonie mobile sont mono (un seul canal de signal pour une restitution sur un seul haut-parleur). Le codec 3GPP EVS (pour « Enhanced Voice Services ») permet d'offrir une qualité « Super-HD » (aussi appelée voix « Haute Définition + » ou HD+) avec une bande audio en bande super-élargie (SWB pour « super-wideband » en anglais) pour des signaux échantillonnés à 32 ou 48 kHz ou pleine bande (FB pour « Fullband ») pour des signaux échantillonnés à 48 kHz ; la largeur de bande audio est de 14,4 à 16 kHz en mode SWB (de 9,6 à 128 kbit/s) et de 20 kHz en mode FB (de 16,4 à 128 kbit/s).

**[0003]** La prochaine évolution de qualité dans les services conversationnels proposés par les opérateurs devrait être constituée par les services immersifs, en utilisant des terminaux tels que des smartphones par exemple équipés de plusieurs microphones ou des équipements de conférence audio spatialisée ou de visioconférence de type télé-présence, ou encore des outils de partage de contenus « live », avec un rendu sonore spatialisé en 3D, autrement plus immersif qu'une simple restitution stéréo 2D. Avec les usages de plus en plus répandus d'écoute sur téléphone mobile avec un casque audio et l'apparition d'équipements audio avancés (accessoires tels qu'un microphone 3D, assistants vocaux avec antennes acoustiques, casques de réalité virtuelle, etc.) et d'outils spécifiques (par exemple de la production de contenu vidéo 360°) la captation et le rendu de scènes sonores spatialisées sont désormais assez communes pour offrir une expérience de communication immersive.

**[0004]** A ce titre, la future norme 3GPP « IVAS » (pour « Immersive Voice And Audio Services ») propose l'extension du codec EVS à l'immersif en acceptant comme format d'entrée du codec au moins les formats de son spatialisé listés ci-dessous (et leurs combinaisons):

- Format multicanal (channel-based en anglais) de type stéréo, 5.1 où chaque canal vient alimenter un haut-parleur (par exemple L et R en stéréo, ou L, R, Ls, Rs et C en 5.1)
- Format objet (object-based en anglais) où des objets sonores sont décrits comme un signal audio (en général mono) associé à des métadonnées décrivant les attributs de cet objet (position dans l'espace, largeur spatiale de la source, etc.), et
- Format ambisonique (scene-based en anglais) qui décrit le champ sonore en un point donné, en général capté par un microphone sphérique ou synthétisé dans le domaine des harmoniques sphériques.

**[0005]** On s'intéresse ci-après typiquement au codage d'un son au format ambisonique, à titre d'exemple de réalisation (au moins certains aspects présentés en lien avec l'invention ci-après pouvant également s'appliquer à d'autres formats que de l'ambisonique).

**[0006]** L'ambisonique est une méthode d'enregistrement (« codage » au sens acoustique) de son spatialisé et un système de reproduction (« décodage » au sens acoustique). Un microphone ambisonique (à l'ordre 1) comprend au moins quatre capsules (typiquement de type cardoïde ou sous-cardoïde) arrangées sur une grille sphérique, par exemple les sommets d'un tétraèdre régulier. Les canaux audio associés à ces capsules s'appellent le « A-format ». Ce format est converti dans un « B-format », dans lequel le champ sonore est décomposé en quatre composantes (harmoniques sphériques) notées W, X, Y, Z, qui correspondent à quatre microphones virtuels coïncidents. La composante W correspond à une captation omnidirectionnelle du champ sonore alors que les composantes X, Y et Z, plus directives, sont assimilables à des gradients de pression orientés suivant les trois dimensions de l'espace. Un système ambisonique est un système flexible dans le sens où l'enregistrement et la restitution sont séparés et découplés. Il permet un décodage (au sens acoustique) sur une configuration quelconque de haut-parleurs (par exemple, binaural, son « surround » de type 5.1 ou périophonie (avec élévation) de type 7.1.4). Bien entendu, l'approche ambisonique peut être généralisée à plus de quatre canaux en B-format et cette représentation généralisée est couramment nommée « HOA » (pour « Higher-Order Ambisonics »). Le fait de décomposer le son sur plus d'harmoniques sphériques améliore la précision spatiale de restitution lors d'un rendu sur hauts-parleurs.

**[0007]** Un signal ambisonique à l'ordre N comprend  $(N+1)^2$  composantes et, à l'ordre 1 (si  $N=1$ ), on retrouve les quatre composantes de l'ambisonique original qui est couramment appelé FOA (pour First-Order Ambisonics). Il existe aussi une variante dite « planaire » de l'ambisonique qui décompose le son défini dans un plan qui est en général le plan horizontal. Dans ce cas, le nombre de composantes est de  $2N+1$  canaux. L'ambisonique d'ordre 1 (4 canaux : W, X, Y, Z) et l'ambisonique d'ordre 1 planaire (3 canaux : W, X, Y) sont désignés ci-après par « ambisonique » indistinctement pour faciliter la lecture, les traitements présentés étant applicables indépendamment du type planaire ou non. Si toutefois dans certains passages il est besoin de faire une distinction, les termes « ambisonique d'ordre 1 » et « ambisonique d'ordre 1 planaire » sont utilisés. On remarquera que l'on peut dériver du B-format à l'ordre 1 un signal stéréo (2 canaux) correspondant à des captations stéréo coïncidentes de type Blumlein Crossed Pair (X+Y et X-Y) ou Mid-Side (en combinant W et X pour le Mid et en prenant Y comme Side).

**[0008]** Par la suite, on appelle « son ambisonique » un signal en B-format à un ordre prédéterminé. Dans des variantes, le son ambisonique peut être défini dans un autre format tel que le A-format ou des canaux précombinés par matriçage fixe (conservant le nombre de canaux ou le réduisant à un cas à 3 ou 2 canaux), comme on le verra plus loin.

**[0009]** Les signaux à traiter par le codeur/décodeur se présentent comme des successions de blocs d'échantillons sonores appelés « trames » ou « sous-trames » ci-après.

**[0010]** En outre, ci-après, les notations mathématiques suivent la convention suivante :

- Vecteur : **u** (minuscule, gras)
- Matrice : **A** (majuscule, gras)

**[0011]** L'approche la plus simple pour coder un signal stéréo ou ambisonique consiste à utiliser un codeur mono et de l'appliquer en parallèle à tous les canaux avec éventuellement une allocation des bits différente selon les canaux. Cette approche est appelée ici « multi-mono » (même si en pratique on peut généraliser l'approche à du multi-stéréo ou une utilisation de plusieurs instances parallèles d'un même codec coeur).

**[0012]** Une telle réalisation est présentée à la figure 1. Le signal d'entrée est divisé en canaux (mono) par le bloc 100. Ces canaux sont codés individuellement par les blocs 120 à 122 en fonction d'une allocation prédéterminée. Leur train binaire est multiplexé (bloc 130) et après transmission et/ou stockage il est démultiplexé (bloc 140) pour appliquer un décodage de chacun des canaux (blocs 150 à 152) qui sont recombinaés (bloc 160).

La qualité associée varie selon le codage mono utilisé, et elle n'est en général satisfaisante qu'à très haut débit, par exemple avec un débit d'au moins 48 kbit/s par canal mono pour un codage EVS. Ainsi à l'ordre 1 on obtient un débit minimal de  $4 \times 48 = 192$  kbit/s.

**[0013]** Les solutions proposées actuellement pour des codecs plus sophistiqués, pour de la spatialisation ambisonique notamment, ne sont pas satisfaisantes, notamment en termes de complexité, retard et utilisation efficace du débit, pour assurer une décorrélation efficace entre canaux ambisoniques.

**[0014]** Par exemple, le codec MPEG-H pour les sons ambisoniques utilise une opération d'addition-recouvrement qui ajoute du retard et de la complexité, ainsi qu'une interpolation linéaire sur des vecteurs de directions qui est sous-optimale et introduit des défauts. Un problème de base de ce codec est qu'il met en oeuvre une décomposition en composantes prédominantes et ambiance car les composantes prédominantes sont censées être perceptuellement distinctes de l'ambiance, mais cette décomposition n'est pas complètement spécifiée. Le codeur MPEG-H souffre de problème de non-correspondance entre les directions des composantes principales d'une trame à l'autre : l'ordre des composantes (signaux) peut être permuté tout comme les directions associées. C'est la raison pour laquelle le codec MPEG-H utilise une technique de « matching » et d'addition-recouvrement (overlap-add en anglais) afin de résoudre ce problème.

**[0015]** Par ailleurs, il serait possible d'utiliser des approches de codage fréquentiel (dans le domaine FFT ou MDCT) plutôt qu'un codage temporel comme dans le codec MPEG-H, mais un traitement des signaux dans le domaine fréquentiel (sous-bandes) oblige à transmettre à un décodeur des données par sous-bande, en augmentant ainsi le débit nécessaire à cette transmission.

**[0016]** La présente invention vient améliorer cette situation.

Elle propose à cet effet un procédé de codage en compression de signaux sonores formant une succession dans le temps de trames d'échantillons, dans chacun de N canaux en représentation ambisonique d'ordre supérieur à 0, le procédé comportant :

- former, à partir des canaux pour une trame courante, une matrice de covariance entre canaux et rechercher des vecteurs propres de la matrice de covariance pour obtenir une matrice de vecteurs propres,
- tester la matrice de vecteurs propres pour vérifier qu'elle représente une rotation dans un espace de dimension N et corriger sinon la matrice de vecteurs propres jusqu'à obtenir une matrice de rotation, pour la trame courante, et
- appliquer ladite matrice de rotation aux signaux des N canaux avant un encodage par canaux séparés desdits signaux.

**[0017]** Ainsi, la présente invention permet d'améliorer une décorrélation entre les N canaux à encoder séparément par la suite. Cet encodage séparé est désigné aussi ci-après « encodage multi-mono ».

**[0018]** Dans une forme de réalisation, le procédé peut comporter en outre :

- coder des paramètres tirés de la matrice de rotation en vue d'une transmission via un réseau. Ces paramètres peuvent être typiquement des valeurs de quaternion et/ou d'angle de rotation et/ou d'angle d'Euler comme on le verra plus loin, ou encore simplement des éléments de cette matrice par exemple.

**[0019]** Dans une forme de réalisation, le procédé peut comporter en outre :

- comparer la matrice de vecteurs propres obtenue pour la trame courante à une matrice de rotation obtenue pour une trame précédant la trame courante, et
- permuter des colonnes de la matrice de vecteurs propres de la trame courante pour assurer une cohérence avec la matrice de rotation de la trame précédente.

5

Une telle réalisation permet de conserver une homogénéité globale et d'éviter notamment des clics audibles d'une trame à l'autre, pendant la restitution sonore.

**[0020]** Toutefois, certaines transformations mises en oeuvre pour l'obtention des vecteurs propres à partir de la matrice de covariance (comme la « PCA/KLT » vue plus loin) sont susceptibles d'inverser le sens de certains des vecteurs propres et il convient alors de vérifier à la fois une cohérence d'axe, puis de direction sur cet axe, de chaque vecteur propre de la matrice de la trame courante. A cet effet, dans une forme de réalisation, la permutation précitée des colonnes permettant d'assurer déjà une cohérence d'axes des vecteurs, le procédé comporte en outre :

10

- vérifier, pour chaque vecteur propre de la trame courante, une cohérence de direction avec un vecteur-colonne de position correspondante de la matrice de rotation de la trame précédente, et
- en cas d'incohérence, inverser le signe des éléments de ce vecteur propre dans la matrice de vecteurs propres de la trame courante.

15

**[0021]** Typiquement, une permutation entre colonnes de la matrice de vecteurs propres inversant le signe d'un déterminant de la matrice de vecteurs propres et le déterminant d'une matrice de rotation étant égal à 1, on peut estimer le déterminant de la matrice de vecteurs propres, et si ce dernier est égal à -1, on peut alors inverser les signes des éléments d'une colonne choisie de la matrice de vecteurs propres, pour que le déterminant soit égal à 1 et former ainsi une matrice de rotation.

20

**[0022]** Dans une réalisation, le procédé peut comporter en outre :

25

- une estimation d'écart entre la matrice de rotation obtenue pour la trame courante et une matrice de rotation obtenue pour une trame précédant la trame courante,
- en fonction de l'écart estimé, déterminer si au moins une interpolation est à opérer entre la matrice de rotation de la trame courante et la matrice de rotation de la trame précédente.

30

Une telle interpolation permet alors de lisser (« moyenner progressivement ») les matrices de rotation appliquées respectivement à la trame précédente et la trame courante et atténuer ainsi un effet de clic audible d'une trame à l'autre à la restitution.

**[0023]** Dans une telle réalisation :

35

- en fonction de l'écart estimé, il est déterminé un nombre d'interpolations à opérer entre la matrice de rotation de la trame courante et la matrice de rotation de la trame précédente,
- la trame courante est découpée en un nombre de sous-trames correspondant au nombre d'interpolations à opérer, et
- on peut coder au moins ce nombre d'interpolations en vue d'une transmission via le réseau précité.

40

**[0024]** Dans une forme de réalisation, la représentation ambisonique est d'ordre 1 et le nombre N de canaux est quatre, et la matrice de rotation de la trame courante est représentée par deux quaternions.

**[0025]** Dans ce mode de réalisation et dans le cas d'une interpolation, chaque interpolation pour une sous-trame courante est une interpolation sphérique linéaire (ou « SLERP »), menée en fonction de l'interpolation de la sous-trame précédant la sous-trame courante et à partir des quaternions de la sous-trame précédente.

45

**[0026]** Par exemple, l'interpolation sphérique linéaire de la sous-trame courante peut être menée pour obtenir les quaternions de la sous-trame courante comme suit :

50

$$Q_{L,interp}(\alpha) = Q_{L,t-1} \frac{\sin(1-\alpha)\Omega_L}{\sin \Omega_L} + Q_{L,t} \frac{\sin \alpha\Omega_L}{\sin \Omega_L}$$

55

$$Q_{R,interp}(\alpha) = Q_{R,t-1} \frac{\sin(1-\alpha)\Omega_R}{\sin \Omega_R} + Q_{R,t} \frac{\sin \alpha\Omega_R}{\sin \Omega_R}$$

Où:

$Q_{L,t-1}$  est l'un des quaternions de la sous-trame précédente t-1,  
 $Q_{R,t-1}$  est l'autre des quaternions de la sous-trame précédente t-1,  
 $Q_{L,t}$  est l'un des quaternions de la sous-trame courante t,  
 $Q_{R,t}$  est l'autre des quaternions de la sous-trame courante t,  
 $\Omega_L = \text{Arccos}(Q_{L,t-1} \cdot Q_{L,t})$ ;  $\Omega_R = \text{Arccos}(Q_{R,t-1} \cdot Q_{R,t})$   
 et  $\alpha$  correspond à un facteur d'interpolation.

**[0027]** Dans une forme de réalisation, la recherche des vecteurs propres est effectuée par analyse en composantes principales (ou « PCA ») ou par transformée de Karhunen Loeve (ou « KLT »), dans le domaine temporel.

Bien entendu, d'autres réalisations peuvent être envisagées (décomposition en valeurs singulières, ou autres).

**[0028]** Dans une forme de réalisation, le procédé comporte une étape préalable de prévision de budget d'allocation de bits par canal ambisonique, comprenant :

- pour chaque canal ambisonique, une estimation d'énergie acoustique courante dans le canal,
- la sélection dans une mémoire d'un score prédéterminé, de qualité, fonction de ce canal ambisonique et d'un débit courant dans le réseau,
- l'estimation d'une pondération à opérer pour l'allocation de bits à ce canal, par multiplication du score sélectionné à l'énergie estimée.

Cette réalisation permet alors de gérer une allocation de bits optimale à attribuer pour chaque canal à coder. Elle est avantageuse en tant que telle et pourrait éventuellement faire l'objet d'une protection séparée.

**[0029]** La présente invention vise aussi un procédé de décodage de signaux sonores formant une succession dans le temps de trames d'échantillons, dans chacun de N canaux en représentation ambisonique d'ordre supérieur à 0, le procédé comportant:

- recevoir, pour une trame courante, en plus des signaux des N canaux de cette trame courante, des paramètres d'une matrice de rotation,
- construire une matrice de rotation inverse à partir desdits paramètres,
- appliquer ladite matrice de rotation inverse à des signaux issus des N canaux reçus, avant un décodage par canaux séparés desdits signaux.

Une telle réalisation permet d'améliorer aussi au décodage une décorrélation entre les N canaux.

**[0030]** La présente invention vise aussi un dispositif de codage comportant un circuit de traitement pour la mise en oeuvre du procédé de codage présenté précédemment.

**[0031]** Elle vise aussi un dispositif de décodage comportant un circuit de traitement pour la mise en oeuvre du procédé de décodage ci-avant.

**[0032]** Elle vise aussi un programme informatique comportant des instructions pour la mise en oeuvre du procédé ci-avant, lorsque ces instructions sont exécutées par un processeur d'un circuit de traitement.

Elle vise aussi un support mémoire non-transitoire stockant les instructions d'un tel programme informatique.

**[0033]** D'autres avantages et caractéristiques de l'invention apparaîtront à la lecture d'exemples de réalisation présentés dans la description détaillée ci-après, et à l'examen des dessins annexés sur lesquels :

- la figure 1 illustre un codage multi-mono (état de l'art),
- la figure 2 illustre une succession d'étapes principales d'un exemple procédé au sens de l'invention,
- la figure 3 présente la structure générale d'un exemple de codeur selon l'invention,
- la figure 4 présente détaille l'analyse et la transformation PCA/KLT réalisée par le bloc 310 du codeur de la figure 3,
- la figure 5 présente un exemple de décodeur selon l'invention,
- la figure 6 présente le décodage et la synthèse PCA/KLT inverse de la figure 4, au décodage,
- la figure 7 illustre des exemples de réalisation structurelle d'un codeur et d'un décodeur au sens de l'invention.

**[0034]** L'invention vise à permettre un codage optimisé par :

- un matriçage adaptatif en temporel (en particulier avec une transformation adaptative obtenue par PCA/KLT (« PCA » désignant une analyse en composante principale et « KLT » désignant une transformée de Karhunen Loeve),
- suivi préférentiellement par un codage multi-mono.

Le matriçage adaptatif permet une décomposition en canaux plus efficaces qu'un matriçage fixe. Le matriçage selon l'invention permet avantageusement de décorréler les canaux avant codage multi-mono, de sorte que le bruit de codage

introduit par le codage de chacun des canaux déforme globalement le moins possible l'image spatiale lorsque les canaux sont recombinaés pour reconstruire un signal ambisonique au décodage.

De plus, l'invention permet d'assurer une adaptation douce des paramètres de matricage afin d'éviter des artéfacts de type « clics » en bordure de trame ou des fluctuations trop rapides d'image spatiale, ou encore des artéfacts de codage dus à des variations trop fortes (par exemple liées à des permutations intempestives de sources sonores entre canaux) dans les différents canaux individuels issus du matricage qui sont ensuite codés par des instances différentes d'un codec mono. Il est présenté ci-après un codage multi-mono avec allocation préférentiellement variable des bits entre canaux (après matricage adaptatif), mais dans des variantes plusieurs instances d'un codec coeur stéréo ou autre peuvent être utilisées.

**[0035]** Afin de faciliter la compréhension de l'invention, il est rappelé ci-après certains concepts explicatifs concernant les rotations en dimension  $n$ , les décompositions de type PCA/KLT ou SVD (« SVD » désignant une décomposition en valeurs singulières).

### Les rotations et les « quaternions »

**[0036]** Les signaux sont représentés par blocs successifs d'échantillons sonores, ces blocs étant appelés « sous-trames » ci-après.

L'invention utilise une représentation des rotations en dimension  $n$  avec des paramètres adaptés pour une quantification par trame et surtout une interpolation efficace par sous-trame. On définit ci-dessous les représentations de rotations utilisées en dimension 2, 3 et 4.

**[0037]** Une rotation (autour de l'origine) est une transformation de l'espace en dimension  $n$  qui modifie un vecteur en un autre vecteur, telle que :

- L'amplitude du vecteur est préservée
- Le produit vectoriel de vecteurs définissant un repère orthonormé avant rotation est préservé après rotation (il n'y a pas de réflexion).

Une matrice  $\mathbf{M}$  de taille  $n \times n$  est une matrice de rotation si et seulement si  $\mathbf{M}^T \cdot \mathbf{M} = \mathbf{I}_n$  où  $\mathbf{I}_n$  désigne la matrice identité de taille  $n \times n$  (c'est-à-dire que  $\mathbf{M}$  est une matrice unitaire,  $\mathbf{M}^T$  désignant la transposée de  $\mathbf{M}$ ) et son déterminant vaut +1.

**[0038]** On utilise dans l'invention plusieurs représentations qui sont équivalentes à la représentation par matrice de rotation :

En deux dimensions (dans un plan 2D) ( $n=2$ ) : On utilise comme représentation l'angle de rotation comme suit.

**[0039]** Etant donné l'angle de rotation  $\theta$  on en déduit la matrice de rotation :

$$\mathbf{M}_2(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix}$$

**[0040]** Etant donnée une matrice de rotation, on peut calculer l'angle  $\theta$  en observant que la trace de la matrice est  $2\cos \theta$ . On notera qu'il est également possible d'estimer  $\theta$  directement à partir d'une matrice de covariance avant d'appliquer une décomposition en composantes principales (PCA) et décomposition en valeurs propres (EVD) présentées plus loin.

**[0041]** L'interpolation entre deux rotations d'angles respectifs  $\theta_1$  et  $\theta_2$  peut se faire par interpolation linéaire entre  $\theta_1$  et  $\theta_2$ , en prenant en compte la contrainte de plus court chemin sur le cercle unité entre ces deux angles.

**[0042]** Dans l'espace en trois dimensions (3D) ( $n=3$ ): On utilise comme représentation les angles d'Euler et les quaternions. Dans des variantes on pourra utiliser également une représentation par axe-angle qui n'est pas rappelée ici.

**[0043]** Une matrice de rotation de taille  $3 \times 3$  peut être décomposée en un produit de 3 rotations élémentaires d'angle  $\theta$  selon les axes  $x$ ,  $y$ , ou  $z$ .

$$\mathbf{M}_{3,x}(\theta) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & -\sin \theta \\ 0 & \sin \theta & \cos \theta \end{pmatrix}$$

$$\mathbf{M}_{3,y}(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}$$

$$M_{3,z}(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

5

[0044] Selon les combinaisons d'axes, les angles sont dits d'Euler ou de Cardan.

[0045] Une autre représentation des rotations 3D toutefois est donnée par les quaternions. Les quaternions sont une généralisation des représentations par nombres complexes avec quatre composantes sous la forme d'un nombre  $q = a + bi + cj + dk$  où  $i^2 = j^2 = k^2 = ijk = -1$ .

10 [0046] La partie réelle  $a$  est appelée scalaire et les trois parties imaginaires ( $b, c, d$ ) forment un vecteur 3D. La norme

d'un quaternion est  $|q| = \sqrt{a^2 + b^2 + c^2 + d^2}$ . Les quaternions unitaires (de norme 1) représentent les rotations - cependant cette représentation n'est pas unique ; ainsi, si  $q$  représente une rotation,  $-q$  représente la même rotation.

15 [0047] Etant donné un quaternion unitaire  $q = a + bi + cj + dk$  (avec  $a^2 + b^2 + c^2 + d^2 = 1$ ), la matrice de rotation associée est :

$$M_{3,quat}(q) = \begin{pmatrix} a^2 - b^2 - c^2 - d^2 & 2(bc - ad) & 2(ac + bd) \\ 2(ad + bc) & a^2 - b^2 + c^2 - d^2 & 2(cd - ab) \\ 2(bd - ac) & 2(ab + cd) & a^2 - b^2 - c^2 + d^2 \end{pmatrix}$$

20

[0048] Les angles d'Euler ne permettent pas d'interpoler correctement des rotations 3D ; pour ce faire on utilise plutôt les quaternions ou la représentation axe-angle. La méthode de l'interpolation SLERP (pour « spherical linear interpolation ») consiste à interpoler selon la formule :

25

$$slerp(q_1, q_2, \alpha) = \frac{\sin(1 - \alpha)\Omega}{\sin \Omega} q_1 + \frac{\sin \alpha\Omega}{\sin \Omega} q_2$$

30

où  $0 \leq \alpha \leq 1$  est le facteur d'interpolation pour aller de  $q_1$  à  $q_2$  et  $\Omega$  est l'angle entre les deux quaternions:

$$\Omega = \arccos(q_1 \cdot q_2)$$

35

où  $q_1 \cdot q_2$  désigne le produit scalaire entre deux quaternions (identique au produit scalaire entre deux vecteurs de dimension 4).

[0049] Cela revient à interpoler en suivant un grand cercle sur une sphère 4D avec une vitesse angulaire constante en fonction de  $\alpha$ . Il convient de s'assurer que le plus court chemin est utilisé pour l'interpolant en changeant le signe de l'un des quaternions quand  $q_1 \cdot q_2 < 0$ . On notera que d'autres méthodes d'interpolation de quaternions peuvent être utilisées (normalized linear interpolation ou nlerp, splines, ...).

40

On remarquera qu'il est également possible d'interpoler des rotations 3D par le biais de la représentation axe-angle ; dans ce cas l'angle est interpolé comme dans le cas 2D et l'axe peut être interpolé par exemple par la méthode SLERP (en 3D) en s'assurant que le plus court chemin est pris sur une sphère unité 3D et en tenant compte du fait que la représentation donnée par l'axe  $r$  et l'angle  $\theta$  est équivalente à celle donnée par l'axe de direction opposée  $-r$  et l'angle  $2\pi - \theta$ .

45

[0050] En dimension 4 ( $n=4$ ), une rotation peut être paramétrée par 6 angles ( $n(n-1)/2$ ) et on montre que la multiplication

de deux matrices de taille 4x4 appelées quaternion ( $Q_1$ ) et anti-quaternion ( $Q_2^*$ ) associées à des quaternions  $q_1 = a + bi + cj + dk$  et  $q_2 = w + xi + yj + zk$  donne une matrice de rotation de taille 4x4.

50

Il est possible de retrouver le double quaternion associé ( $q_1, q_2$ ) et des matrices de quaternion et anti-quaternion associées telles que :

55

$$Q_1 = \begin{pmatrix} a & b & c & d \\ -b & a & -d & c \\ -c & d & a & -b \\ -d & -c & b & a \end{pmatrix}$$

et

$$Q_2^* = \begin{pmatrix} w & -x & -y & -z \\ x & w & -z & y \\ y & z & w & -x \\ z & -y & x & w \end{pmatrix}$$

[0051] Leur produit donne une matrice de taille 4x4 :

$$M_{4,quat}(q_1, q_2) = Q_1 Q_2^*$$

et il est possible de vérifier que cette matrice vérifie les propriétés d'une matrice de rotation (matrice unitaire et déterminant égal à 1).

[0052] Inversement, étant donné une matrice de rotation 4x4, on peut factoriser cette matrice en un produit de matrices

sous la forme  $Q_1 Q_2^*$ , par exemple avec la méthode dite « factorisation de Cayley ». Cela implique de calculer une matrice intermédiaire appelée « transformée tétragonale » (ou matrice associée) et d'en déduire les quaternions à une indétermination près sur le signe des deux quaternions (qui peut être levée par une contrainte supplémentaire de « plus court chemin » évoquée plus loin).

#### Décomposition en valeurs singulières (ou « SVD »)

[0053] La décomposition en valeurs singulières (singular value decomposition ou SVD en anglais) consiste à factoriser une matrice réelle  $A$  de taille  $m \times n$  sous la forme :

$$A = U \Sigma V^T$$

où  $U$  est une matrice unitaire ( $U^T U = I_m$ ) de taille  $m \times m$ ,  $\Sigma$  est une matrice diagonale rectangulaire de taille  $m \times n$  à coefficients réels et positifs  $\sigma_i \geq 0$  ( $i = 1 \dots p$  où  $p = \min(m, n)$ ),  $V$  est une matrice unitaire ( $V^T V = I_n$ ) de taille  $n \times n$  et  $V^T$  est la transposée de  $V$ . Les coefficients  $\sigma_i$  dans la diagonale de  $\Sigma$  sont les valeurs singulières de la matrice  $A$ . Par convention, elles sont en général listées par ordre décroissant, et dans ce cas la matrice diagonale  $\Sigma$  associée à  $A$  est unique.

[0054] Le rang  $r$  de  $A$  est donné par le nombre de coefficients  $\sigma_i$  non nuls. On peut donc réécrire la décomposition en valeurs singulières comme:

$$A = [U_r \tilde{U}_r] \begin{bmatrix} \Sigma_r & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_r^T \\ \tilde{V}_r^T \end{bmatrix}$$

où  $U_r = [u_1, u_2, \dots, u_r]$  sont les vecteurs singuliers à gauche (ou vecteurs de sortie) de  $A$ ,  $\Sigma_r = \text{diag}(\sigma_1, \dots, \sigma_r)$  and  $V_r = [v_1, v_2, \dots, v_r]$  sont les vecteurs singuliers à droite (ou vecteurs d'entrée) de  $A$ . Cette formulation matricielle peut être aussi ré-écrite comme:

$$A = \sum_{i=1}^r \sigma_i u_i v_i^T$$

### EP 3 706 119 A1

**[0055]** Si la somme est limitée à un indice  $i < r$  on obtient une matrice « filtrée » qui ne représente que l'information « prépondérante ».

On peut aussi écrire :

5

$$\mathbf{A} \mathbf{v}_i = \sigma_i \mathbf{u}_i$$

**[0056]** Qui montre que la matrice  $\mathbf{A}$  transforme  $\mathbf{v}_i$  en  $\sigma_i \mathbf{u}_i$ .

**[0057]** La SVD de  $\mathbf{A}$  a une relation avec la décomposition en valeurs propres de  $\mathbf{A}^T \mathbf{A}$  et  $\mathbf{A} \mathbf{A}^T$  car :

10

$$\mathbf{A}^T \mathbf{A} = \mathbf{V} (\boldsymbol{\Sigma}^T \boldsymbol{\Sigma}) \mathbf{V}^T$$

15

$$\mathbf{A} \mathbf{A}^T = \mathbf{U} (\boldsymbol{\Sigma} \boldsymbol{\Sigma}^T) \mathbf{U}^T$$

**[0058]** Les valeurs propres de  $\boldsymbol{\Sigma}^T \boldsymbol{\Sigma}$  et  $\boldsymbol{\Sigma} \boldsymbol{\Sigma}^T$  sont  $\sigma_1^2, \dots, \sigma_r^2$ . Les colonnes de  $\mathbf{U}$  sont les vecteurs propres de  $\mathbf{A} \mathbf{A}^T$ , tandis que les colonnes de  $\mathbf{V}$  sont les vecteurs propres de  $\mathbf{A}^T \mathbf{A}$ .

20

**[0059]** La SVD peut être interprétée de façon géométrique : l'image d'une sphère en dimension  $n$  par la matrice  $\mathbf{A}$  est en dimension  $m$  une hyper-ellipse ayant des axes principaux selon les directions  $\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_m$  et de longueur  $\sigma_1, \dots, \sigma_m$ .

**Transformée de Karhunen Loeve (ou « KLT » pour « Karhunen Loeve Transform »)**

25

**[0060]** La transformation de Karhunen Loeve (KLT) d'un vecteur aléatoire  $\mathbf{x}$  centré en 0 et de matrice de covariance  $\mathbf{R}_{\mathbf{xx}} = \mathbf{E} [\mathbf{x} \mathbf{x}^T]$  est définie par:

30

$$\mathbf{y} = \mathbf{V}^T \mathbf{x}$$

où  $\mathbf{V}$  est la matrice de vecteurs propres (avec la convention que les vecteurs propres sont des vecteurs colonne) obtenue par décomposition en valeurs propres de  $\mathbf{R}_{\mathbf{xx}}$

35

$$\mathbf{R}_{\mathbf{xx}} = \mathbf{V} \boldsymbol{\Lambda} \mathbf{V}^T$$

où  $\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \dots, \lambda_n)$  est une matrice diagonale dont les coefficients sont les valeurs propres. La matrice  $\mathbf{V} = [\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n]$  contient les vecteurs propres (colonnes) de  $\mathbf{R}_{\mathbf{xx}}$ , tels que

40

$$\mathbf{R}_{\mathbf{xx}} \mathbf{v}_i = \lambda_i \mathbf{v}_i$$

**[0061]** On peut voir la KLT comme un changement de base, car le produit  $\mathbf{V}^T \mathbf{x}$  exprime le vecteur  $\mathbf{x}$  dans la base donnée par les vecteurs propres.

45

La transformation inverse est donnée par:

$$\mathbf{x} = \mathbf{V} \mathbf{y}$$

50

**[0062]** La KLT permet de décorrélérer les composantes de  $\mathbf{x}$  ; les variances du vecteur transformé  $\mathbf{y}$  sont les valeurs propres de  $\mathbf{R}_{\mathbf{xx}}$ .

**Analyse en composantes principales (ou « PCA » pour « principal component analysis »)**

55

**[0063]** L'analyse en composante principale (PCA) est une technique de réduction de dimensionnalité qui produit des variables orthogonales et maximise la variance des variables après projection (ou de façon équivalente minimiser l'erreur de reconstruction).

**[0064]** La PCA présentée ci-après, bien que s'appuyant aussi sur une décomposition en valeurs propres comme la KLT, est telle que la matrice de covariance estimée  $\hat{R}_{xx}$  est calculée à partir de N vecteurs observés  $x_i$ ,  $i = 1 \dots N$  de dimension n:

5

$$\hat{R}_{xx} = \frac{1}{N-1} \sum_{i=1}^N x_i x_i^T$$

10 en supposant que ces vecteurs sont centrés :

15

$$m_x = \frac{1}{N} \sum_{i=1}^N x_i = 0$$

**[0065]** La décomposition en valeurs propres de  $\hat{R}_{xx}$  sous la forme  $\hat{R}_{xx} = V \Lambda V^T$  permet de calculer les composantes principales:  $y_n = V^T x_n$ .

20

La PCA est une transformation par la matrice  $V^T$  qui projette les données dans une nouvelle base pour maximiser la variance des variables après projection.

On notera que la PCA peut également s'obtenir à partir d'une SVD du signal  $x_i$  mis sous la forme d'une matrice  $X$  de taille  $n \times N$ . Dans ce cas, on peut écrire :

25

$$X = U D V^T$$

**[0066]** On vérifie que  $XX^T = U D D^T U^T$  qui correspond à une diagonalisation de  $XX^T$ . Ainsi les vecteurs de projection de la PCA correspondent aux vecteurs colonne de  $U$  et la projection donne comme résultat  $U^T X = D V^T$ .

30

**[0067]** On notera également que la PCA est en général vue comme une technique de réduction de dimensionnalité, pour « compresser » un jeu de données en grande dimension vers un jeu comprenant peu de composantes principales. Dans l'invention, la PCA permet avantageusement de décorrélérer le signal multidimensionnel en entrée mais on évite de supprimer des canaux (donc réduire le nombre de canaux) pour éviter d'introduire des artefacts. On force ainsi un débit de codage minimal pour éviter de « tronquer » l'image spatiale, sauf dans des variantes spécifiques où des valeurs propres sont tellement faibles qu'un débit nul peut être autorisé (par exemple pour mieux coder des sons ambisoniques créés artificiellement avec une seule source spatialisée de façon synthétique).

35

**[0068]** On se réfère maintenant à la figure 2 pour décrire des principes généraux des étapes qui sont mises en oeuvre dans un procédé au sens de l'invention, pour une trame courante  $t$ .

L'étape S1 consiste à obtenir les signaux respectifs des canaux ambisoniques (ici quatre canaux W, Y, Z, X dans l'exemple décrit utilisant un ordre de canaux selon la convention ACN pour Ambisonics Channel Number), pour chaque trame  $t$ . Ces signaux peuvent être mis sous la forme d'une matrice  $n \times L$  (pour  $n$  canaux ambisoniques (ici 4) et  $L$  échantillons par trame).

40

A l'étape suivante S2, on peut optionnellement pré-traiter les signaux de ces canaux par exemple par un filtre passe-haut comme décrit plus loin en référence à la figure 3.

A l'étape suivante S3, on applique à ces signaux une analyse en composantes principales PCA ou de façon équivalente une transformée de Karhunen Loeve KLT, pour obtenir des valeurs propres et une matrice de vecteurs propres à partir d'une matrice de covariance des  $n$  canaux. Dans des variantes de l'invention une SVD pourra être utilisée.

45

A l'étape S4, cette matrice de vecteurs propres, obtenue pour la trame courante  $t$ , subit des permutations signées pour qu'elle soit la plus alignée possible avec la matrice de même nature de la trame précédente  $t-1$ . Dans le principe, on s'assure que l'axe des vecteurs colonnes dans la matrice de vecteurs propres correspond le plus possible à l'axe des vecteurs colonnes à la même place dans la matrice de la trame précédente et sinon, on permute les positions des vecteurs propres de la matrice de la trame courante  $t$  qui ne correspondent pas. Ensuite, on s'assure en outre que les directions des vecteurs propres d'une matrice à l'autre coïncident également. En d'autres termes, on ne s'intéresse dans un premier temps qu'aux droites qui portent les vecteurs propres (juste la direction, sans le sens) et on cherche pour chaque droite la droite la plus proche dans la matrice de la trame précédente  $t-1$ . Pour cela on permute des vecteurs dans la matrice de la trame courante. Puis dans un second temps, on cherche à faire correspondre l'orientation des vecteurs (sens). Pour cela, on inverse le signe des vecteurs propres qui n'auraient pas le bon sens. Une telle réalisation permet d'assurer une cohérence maximale entre les deux matrices et éviter ainsi des clics audibles entre deux trames lors d'une restitution sonore.

55

A l'étape S5, on s'assure en outre que la matrice de vecteurs propres de la trame courante  $t$ , ainsi corrigée par permutations signées, représente bien l'application d'une rotation (d'un angle pour  $n=2$  canaux, de trois angles d'Euler, d'un axe et d'un angle ou d'un quaternion pour  $n=3$  correspondant à la représentation ambisonique d'ordre 1 planaire W, Y, Z, et de deux quaternions pour  $n=4$  en représentation ambisonique d'ordre 1 de type W,Y,Z,X).

Pour s'assurer qu'il s'agit bien d'une rotation, le déterminant de la matrice de vecteurs propres de la trame courante  $t$ , corrigée par permutations, doit être positif et égal à (ou, en pratique, voisin de) +1 à l'étape S6. S'il est égal à (ou proche de) -1, alors il convient de :

- 10 - permuter à nouveau deux vecteurs propres (par exemple associés à des canaux de faible énergie, donc peu représentatifs), ou
- préférentiellement d'inverser le signe de tous les éléments d'une colonne (par exemple associée à un canal de faible énergie) à l'étape S6.

15 On obtient alors une matrice de vecteurs propres pour la trame courante  $t$  correspondant effectivement à une rotation à l'étape S7.

On peut alors coder sur un nombre de bits alloués à cet effet des paramètres de cette matrice (comme par exemple la valeur d'angle, d'un axe et d'un angle, ou de quaternion(s) de cette matrice) à l'étape S8. Dans une autre réalisation optionnelle mais avantageuse, dans le cas où il est constaté à l'étape S9 un écart significatif (supérieur à un seuil par exemple) entre la matrice de rotation estimée pour la trame courante  $t$  et la matrice de rotation de la trame précédente  $t-1$ , on peut déterminer un nombre variables de sous-frames d'interpolation : autrement on fixe ce nombre de sous-frames à une valeur pré-déterminée. L'étape S10 consiste à :

- découper la trame courante en sous-frames, et
- 25 - interpoler des matrices à appliquer aux sous-frames successives depuis la matrice de la trame précédente  $t-1$  jusqu'à la matrice de la trame courante  $t$ , afin de lisser dans le temps la différence entre les deux matrices.

A l'étape S11, on applique les matrices de rotation interpolées à une matrice  $n \times (L/K)$  représentant chacune des  $K$  sous-frames des signaux des canaux ambisoniques de l'étape S1 (ou optionnellement S2) pour décorrélérer autant que possible ces signaux avant l'encodage multi-mono de l'étape S14. Il est rappelé en effet qu'il est souhaité dé-corrélérer autant que possible ces signaux avant cette transformation multi-mono, selon une approche générale. Une allocation binaire aux canaux séparés est faite à l'étape S12 et codée à l'étape S13.

A l'étape S14, avant d'opérer le multiplexage de l'étape S15 et finir ainsi le procédé de codage en compression, on peut décider d'un nombre de bits à allouer par canal en fonction de la représentativité de ce canal et du débit disponible sur le réseau RES (figure 7). Dans une forme de réalisation, on estime l'énergie dans chaque canal pour une trame courante et on multiplie cette énergie par un score prédéfini pour ce canal et pour un débit donné (ce score étant par exemple une note MOS explicitée plus loin en référence à la figure 3). On pondère ainsi le nombre de bits à allouer pour chaque canal. Une telle réalisation est avantageuse en tant que tel et peut éventuellement faire l'objet d'une protection séparée en contexte ambisonique.

40 **[0069]** On a illustré sur la figure 7 un dispositif de codage DCOD et un dispositif de décodage DDEC, au sens de l'invention, ces dispositifs étant duals l'un de l'autre (dans le sens de « réversibles ») et reliés l'un à l'autre par un réseau de communication RES.

**[0070]** Le dispositif de codage DCOD comporte un circuit de traitement incluant typiquement :

- 45 - une mémoire MEM1 pour stocker des données d'instructions d'un programme informatique au sens de l'invention (ces instructions pouvant être réparties entre le codeur DCOD et le décodeur DDEC) ;
- une interface INT1 de réception de signaux ambisoniques répartis sur différents canaux (par exemple quatre canaux W, Y, Z, X à l'ordre 1) en vue de leur codage en compression au sens de l'invention ;
- un processeur PROC1 pour recevoir ces signaux et les traiter en exécutant les instructions de programme informatique que stocke la mémoire MEM1, en vue de leur codage ; et
- 50 - une interface de communication COM 1 pour transmettre les signaux codés via le réseau.

**[0071]** Le dispositif de décodage DDEC comporte un circuit de traitement propre, incluant typiquement :

- 55 - une mémoire MEM2 pour stocker des données d'instructions d'un programme informatique au sens de l'invention (ces instructions pouvant être réparties entre le codeur DCOD et le décodeur DDEC comme indiqué précédemment) ;
- une interface COM2 pour recevoir du réseau RES les signaux codés en vue de leur décodage en compression au sens de l'invention ;

- un processeur PROC2 pour traiter ces signaux en exécutant les instructions de programme informatique que stocke la mémoire MEM2, en vue de leur décodage ; et
- une interface de sortie INT2 pour délivrer les signaux décodés sous forme de canaux ambisoniques  $W'$ ,  $Y'$ ,  $Z'$ ,  $X'$ , par exemple en vue de leur restitution.

5

**[0072]** Bien entendu, cette figure 7 illustre un exemple d'une réalisation structurelle d'un codec (codeur ou décodeur) au sens de l'invention. Les figures 3 à 6 commentées plus loin décrivent en détails des réalisations plutôt fonctionnelles de ces codecs.

**[0073]** On se réfère maintenant à la figure 3 pour décrire un dispositif codeur au sens de l'invention.

10 **[0074]** La stratégie du codeur est de dé-corréler au maximum les canaux du signal ambisonique et de les coder avec un codec coeur. Cette stratégie permet de limiter les artéfacts dans le signal ambisonique décodé. Plus particulièrement, on cherche à appliquer une décorrélation optimisée des canaux d'entrée avant un codage multi-mono ici. Par ailleurs, une interpolation dont le coût de calcul pour le codeur et le décodeur est limité car celle-ci est réalisée dans un domaine spécifique (angle en 2D, quaternion en 3D, double quaternion en 4D) permet d'interpoler les matrices de covariance calculées pour l'analyse PCA/KLT plutôt que de répéter plusieurs fois par trame une décomposition en valeurs propres et vecteurs propres.

15 **[0075]** Néanmoins, avant d'aborder le codage coeur opéré au sens de l'invention, il est présenté ici quelques fonctionnalités du codeur qui sont avantageuses comme notamment l'optimisation du budget de bits alloués au codage en fonction de critères perceptifs, vue plus loin.

20 **[0076]** Dans le mode de réalisation décrit ici du codeur, ce dernier peut être typiquement une extension du codeur normalisé 3GPP EVS (pour « Enhanced Voiced Services »). Avantageusement, on peut reprendre les débits de codage EVS sans modifier alors la structure du train binaire EVS. Ainsi, le codage multi-mono (bloc 340 de la figure 3 décrit plus loin) fonctionne ici avec une allocation possible à chaque canal transformé, restreinte aux débits suivants pour un codage en bande audio super-élargie : 9,6 ; 13,2 ; 16,4 ; 24,4 ; 32 ; 48 ; 64 ; 96 et 128 kbit/s.

25 Bien entendu, il est possible d'ajouter des débits supplémentaires (pour avoir une granularité d'allocation plus fine) en modifiant le codec EVS. On peut utiliser aussi un autre codec que de type EVS, par exemple le codec OPUS®.

30 **[0077]** De manière générale, on retient que plus la granularité de codage est fine, et plus il faut réserver de bits pour représenter les combinaisons de débits possibles. Un compromis entre finesse d'allocation et information supplémentaire décrivant l'allocation binaire doit être opéré. Cette allocation est optimisée ici par le bloc 320 de la figure 3, qui est décrit plus loin. Il s'agit d'une caractéristique avantageuse en tant que telle et indépendante de la décomposition en vecteurs propres en vue d'établir une matrice de rotation au sens de l'invention. A ce titre, l'allocation de bits qu'opère le bloc 320 peut faire l'objet d'une protection séparée.

35 **[0078]** En référence à la figure 3, le bloc 300 reçoit un signal d'entrée  $\mathbf{Y}$  dans la trame courante d'indice  $t$ . L'indice  $n$  n'est pas indiqué ici pour ne pas alourdir les notations. Il s'agit d'une matrice de taille  $n \times L$ . Dans une réalisation adaptée en contexte ambisonique d'ordre 1, on a  $n=4$  canaux  $W$ ,  $Y$ ,  $Z$ ,  $X$  (définis ainsi selon l'ordre ACN) qui peuvent être normalisés selon la convention SN3D. Dans une variante, l'ordre des canaux peut être alternativement par exemple  $W$ ,  $X$ ,  $Y$ ,  $Z$  (en suivant la convention FuMA) et la normalisation peut être différente (N3D ou FuMa). Ainsi les canaux  $W$ ,  $Y$ ,  $Z$ ,  $X$  correspondent aux lignes successives :  $y_{1,l}, y_{2,l}, y_{3,l}, y_{4,l}$ , qui seront notées sous la forme de signaux unidimensionnels  $y_l(l)$ ,  $l = 1, \dots, L$ . Il s'agit donc d'une succession d'échantillons de 1 à  $L$  occupant la trame  $t$ .

40 **[0079]** On suppose que le signal (dans chaque canal) est échantillonné à 48 kHz, sans perte de généralité. La longueur de trame est fixée à 20 ms, soit  $L = 960$  échantillons successifs, sans perte de généralité. Dans des variantes on pourra par exemple utiliser une longueur de trame de  $L = 640$  échantillons pour un échantillonnage à 32 kHz.

45 **[0080]** L'analyse PCA/KLT et la transformation PCA/KLT qui sont décrites plus loin sont effectuées dans le domaine temporel. On comprend ainsi que l'on reste ici dans le domaine temporel sans besoin nécessairement d'opérer une transformée en sous-bandes ou plus généralement fréquentielle.

50 **[0081]** A chaque trame, le bloc 300 du codeur applique un prétraitement (optionnel) pour obtenir le signal d'entrée prétraité noté  $\mathbf{Y}$ . Il peut s'agir d'un filtrage passe-haut (de fréquence de coupure typiquement à 20Hz) de chaque nouvelle trame de 20 ms des canaux du signal d'entrée. Cette opération permet d'enlever la composante continue susceptible de biaiser l'estimation de la matrice de covariance de sorte qu'en sortie du bloc 300 le signal peut être considéré comme étant à moyenne nulle. La fonction de transfert est notée  $H_{pre}(z)$ , ainsi on a pour chaque canal :  $X_l(z) = H_{pre}(z)Y_l(z)$ . Si le bloc 300 n'est pas mis en oeuvre on a  $\mathbf{X} = \mathbf{Y}$ . On peut mettre en oeuvre aussi un filtre passe-bas du bloc 340 pour effectuer le codage multi-mono mais lorsque le bloc 300 est mis en oeuvre, le filtrage passe-haut en prétraitement du codage mono qui peut être utilisé dans le bloc 340 est de façon préférentielle désactivé pour éviter de répéter le même prétraitement et réduire ainsi la complexité globale.

55 **[0082]** La fonction de transfert notée  $H_{pre}(z)$  ci-dessus peut être du type :

$$H_{pre}(z) = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 - a_1 z^{-1} - a_2 z^{-2}}$$

5

en appliquant ce filtre à chacun des  $n$  canaux du signal d'entrée dont les coefficients peuvent être tels que présentés dans le tableau ci-dessous :

10

	8 kHz	16 kHz	32 kHz	48 kHz
$b_0$	0.988954248067140	0.994461788958.195	0.997227049904470	0.998150511190452
$b_1$	-1.977908496134280	-1.988923577916390	-1.994454099808940	-1.996301022380904
$b_2$	0.988954248067140	0.994461788958195	0.997227049904470	0.998150511190452
$a_1$	1.977786483776764	1.988892905899653	1.994446410541927	1.996297601769122
$a_2$	-0.978030508491796	-0.988954249933127	-0.994461789075954	-0.996304442992686

15

20

**[0083]** En variante, on peut utiliser un autre type de filtre, par exemple un filtre de Butterworth d'ordre 6 avec une fréquence à 50 Hz.

Dans des variantes, le pré-traitement pourra inclure une étape de matricage fixe qui pourra garder le même nombre de canaux ou réduire le nombre de canaux. Un exemple de matricage appliqué aux quatre canaux d'un signal ambisonique en B-format est donné ci-dessous :

25

$$M_{B \rightarrow A} = \begin{bmatrix} 1/2 & \frac{1}{\sqrt{6}} & 0 & \frac{1}{\sqrt{12}} \\ 1/2 & \frac{-1}{\sqrt{6}} & 0 & \frac{1}{\sqrt{12}} \\ 1/2 & 0 & \frac{1}{\sqrt{6}} & \frac{-1}{\sqrt{12}} \\ 1/2 & 0 & \frac{-1}{\sqrt{6}} & \frac{-1}{\sqrt{12}} \end{bmatrix}$$

30

35

**[0084]** On notera que dans ce cas ce prétraitement devra être inversé au décodage en appliquant un matricage par  $M_{A \rightarrow B} = M_{B \rightarrow A}^{-1}$  du signal décodé pour retrouver les canaux au format d'origine.

40

**[0085]** Le bloc suivant 310 estime à chaque trame  $t$  une matrice de transformation obtenue par détermination des vecteurs propres par PCA/KLT et vérification que la matrice de transformation que forment ces vecteurs propres caractérise bien une rotation. Des précisions quant à l'opération de bloc 310 sont données plus loin en référence à la figure 4. Cette matrice de transformation effectue un matricage des canaux pour les dé-corréler permettant d'appliquer un codage indépendant de type multi-mono par le bloc 340. Comme détaillé plus loin, le bloc 310 transmet au multiplexeur des indices de quantification représentant la matrice de transformation et de façon optionnelle des informations codant le nombre d'interpolations de la matrice de transformation, par sous-trame de la trame courante  $t$ , comme détaillé plus loin également.

45

**[0086]** Le bloc 320 détermine l'allocation de débit optimale pour chaque canal (après transformation PCA/KLT) en fonction d'un budget de bits  $B$  donné. Ce bloc cherche une répartition du débit entre canaux en calculant un score pour chaque combinaison possible de débits ; l'allocation optimale est trouvée en cherchant la combinaison maximisant ce score.

50

Plusieurs critères peuvent être utilisés pour définir un score pour chaque combinaison.

Par exemple, le nombre de débits possibles pour le codage mono d'un canal peut être limité aux neuf débits discrets du codec EVS ayant une bande audio super-élargie : 9,6 ; 13,2 ; 16,4 ; 24,4 ; 32 ; 48 ; 64 ; 96 et 128 kbit/s. Cependant si le codec selon l'invention fonctionne à un débit donné associé à un budget de  $B$  bits dans la trame courante d'indice  $t$ , en général seul un sous-ensemble de ces débits listés est utilisable. Par exemple si le débit du codec est fixé à  $4 \times 13,2 = 52,8$  kbits/s pour représenter quatre canaux et si chaque canal reçoit un budget minimal de 9,6 kbit/s pour garantir une bande super-élargie pour chacun des canaux, les combinaisons possibles de débits pour le codage de canaux

55

séparés doivent respecter la contrainte que le débit utilisé reste inférieur au débit disponible qui correspond à :

$$B_{multimono} = B - B_{overhead},$$

5

où  $B_{overhead}$  correspond au budget de bits pour l'information supplémentaire codée par trame (allocation binaire + données de rotation) comme décrit plus loin. Par exemple,  $B_{overhead}$  peut être de l'ordre de  $B_{overhead} = 55$  bits par trame de 20 ms (soit 2,75 kbit/s) pour le cas d'un codage ambisonique à quatre canaux ; cela comprend 51 bits pour coder la matrice de rotation et 4 bits (comme décrit ci-dessous) pour coder l'allocation des bits pour le codage des canaux séparés. Pour un débit global de  $4 \times 13.2 = 52,8$  kbits/s, cela laisse donc un budget de  $B_{multimono} = 50.05$  kbit/s.

10

**[0087]** Cela donne en termes de débits par canal les permutations de débit par canal suivantes :

- Singleton (9.6, 9.6, 9.6, 9.6) - total = 38.4
- Permutations de (13.2, 9.6, 9.6, 9.6) - total = 42 kbit/s
- 15 - Permutations de (13.2, 13.2, 9.6, 9.6) - total = 45.6 kbit/s
- Permutations de (13.2, 13.2, 13.2, 9.6) - total = 49.2 kbit/s
- Permutations de (16.4, 9.6, 9.6, 9.6) - total = 45.2 kbit/s
- Permutations de (16.4, 13.2, 9.6, 9.6) - total = 48.8 kbit/s

20

On observe que certaines combinaisons respectant la limite de budget maximal ont un débit très inférieur aux autres, et finalement seules deux combinaisons pertinentes peuvent être retenues :

- Permutations de (13.2, 13.2, 13.2, 9.6) - 4 cas et débit non utilisé de  $50.5 - 49.2 = 1.3$  kbit/s
- et Permutations de (16.4, 13.2, 9.6, 9.6) - 12 cas et débit non utilisé de  $50.5 - 48.8 = 1.7$  kbit/s

25

Cela permet d'illustrer que seize combinaisons sont intéressantes en particulier et peuvent être codées sur 4 bits (16 valeurs). Par ailleurs un certain nombre de bits restent potentiellement inutilisés en fonction de l'allocation choisie.

**[0088]** On observe que le codage du maillage adaptatif basé sur un traitement PCA/KLT et en autorisant une allocation binaire flexible peut avoir pour effet des bits non utilisés et, pour certains canaux, un débit inférieur (par exemple 9,6 kbit/s) au débit également réparti entre chacun des canaux (par exemple 13,2 kbit/s par canal).

30

Pour améliorer cette situation, le bloc 320 peut alors évaluer toutes les combinaisons possibles (pertinentes) de débits pour les 4 canaux issus de la transformation PCA/KLT (en sortie du bloc 310) et leur attribuer un score. Ce score est calculé en se basant sur :

35

- l'énergie de chaque canal, et
- une note moyenne qui peut être pré-mémorisée et issue de tests subjectifs ou objectifs, laquelle note, notée MOS (pour « Mean Opinion Score », s'agissant d'une note moyenne sur un panel de testeurs), est associée au débit alloué.

Ce score peut alors être défini par l'équation

40

$$S(b_{t,1}, \dots, b_{t,n}) = \sum_{i=1}^n Q(b_{t,i}) \cdot E_i$$

45

où  $E_i$  est l'énergie dans la trame courante (d'indice  $t$ ) du signal  $s(l)$ ,  $l = \dots, L - 1$  sur le canal  $i$ , avec :

50

$$E_i = \sum_{l=0}^{L-1} s^2(l)$$

**[0089]** L'allocation optimale peut être telle que :

55

$$b_{t,1}^{opt}, \dots, b_{t,n}^{opt} = \arg \max_{b_{t,1}, \dots, b_{t,n} | \sum_{i=1}^n b_{t,i} \leq B} S(b_{t,1}, \dots, b_{t,n})$$

[0090] En variante, le facteur  $E_i$  peut être fixé à la valeur que prend la valeur propre associée au canal  $i$  issue de la décomposition en valeurs propres du signal en entrée du bloc 310 et après permutation signée éventuelle.

[0091] La note MOS  $Q(b_i)$  est de façon préférentielle la note de qualité subjective du codec utilisé pour le codage multi-mono dans le bloc 340 pour un budget  $b_i$  (en nombres de bits) par trame de 20 ms correspondant à un débit  $R_i = 50 b_i$  (en bits/sec). On peut utiliser au départ les notes MOS subjectives (moyennes) d'un codeur normalisé EVS données par :

$\kappa_i$	0	1	2	3	4	5	6	7	8
$b_i$	192	264	328	488	640	960	1280	1920	2560
$R_i$	9600	13200	16400	24400	32000	48000	64000	96000	128000
$Q(b_i)$	3.62	3.79	4.25	4.60	4.53	4.82	4.83	4.85	4.87

[0092] Alternativement, d'autres valeurs de notes MOS pour chacun des débits listés peuvent être issues d'autres tests (subjectifs ou objectifs) prédisant la qualité du codec. Il est aussi possible d'adapter les notes MOS utilisées dans la trame courante en fonction d'une classification du type de signal (par exemple un signal de parole sans bruit de fond, ou parole avec bruit ambiant, ou musique ou contenu mixte), en réutilisant des méthodes de classification que met en oeuvre le codec EVS et en les appliquant au canal  $W$  du signal ambisonique en entrée avant d'effectuer l'allocation binaire. La note MOS peut aussi correspondre à une note moyenne issue de différents types de méthodologies et d'échelles de notation : MOS (absolu) de 1 à 5, DMOS (de 1 à 5), MUSHRA (de 0 à 100).

[0093] Dans une variante où le codeur EVS est remplacé par un autre codec, la liste de débits  $b_i$  et les notes  $Q(b_i)$  peuvent être remplacées en fonction de cet autre codec. On peut également ajouter des débits supplémentaires de codage au codeur EVS et donc compléter la liste de débits et de notes MOS, ou encore modifier le codeur EVS et potentiellement les notes MOS associées.

[0094] Alternativement encore, l'allocation entre les canaux est affinée en pondérant l'énergie par une puissance  $\alpha$  où  $\alpha$  prend une valeur entre 0 et 1. En faisant varier la valeur de  $\alpha$ , on peut ainsi contrôler l'influence de l'énergie dans l'allocation : plus  $\alpha$  est proche de 1 plus l'énergie a de l'importance dans le score, et donc plus l'allocation est inégale entre les canaux. A l'inverse, plus  $\alpha$  est proche de 0 moins l'énergie a de l'importance et plus l'allocation est équi-répartie entre les canaux. Le score est donc exprimé sous la forme :

$$S(b_{t,1}, \dots, b_{t,n}) = \sum_{i=1}^n Q(b_{t,i}) \cdot E_i^\alpha$$

[0095] Alternativement encore, pour rendre l'allocation plus stable, une seconde pondération peut être ajoutée à la fonction de score pour pénaliser les changements de débits inter-trames. Une pénalité est ajoutée au score si la combinaison de débit n'est pas la même dans la trame  $t$  que dans la trame  $t - 1$ . Le score s'exprime alors sous la forme :

$$S(b_{t,1}, \dots, b_{t,n}) = \sum_{i=1}^n Q(b_{t,i}) \cdot E_i^\alpha \cdot (1 + \beta_i)$$

où  $\beta_i$  a pour valeur une constante pré-déterminée (par exemple 0.1) quand  $b_{t,i} = b_{t-1,i}$  et  $\beta_i = 0$  quand  $b_{t,i} \neq b_{t-1,i}$ . Cette pondération supplémentaire permet de limiter les fluctuations trop fréquentes de débit entre les canaux. Avec cette pondération, seuls les changements significatifs d'énergie entraînent un changement de débit. On peut en outre varier la valeur de la constante pour régler une stabilité de l'allocation.

[0096] En référence à nouveau à la figure 3, une fois calculé le débit pour chaque trame, ce débit est codé par le bloc 330 par exemple de façon exhaustive pour toutes les combinaisons de débits. Dans le cas de 9 débits et 4 canaux, le débit nécessaire est de  $\lceil \log_2(9^4) \rceil = 13$  bits, où  $\lceil \cdot \rceil$  correspond à l'arrondi à l'entier supérieur. La combinaison des 4 débits

peut être codée sous la forme de l'indice :  $\sum_{i=1}^n 9^i \kappa_i$ . Cependant on peut préférer énumérer (au départ, hors ligne) les différentes combinaisons de débits pertinentes pour le budget de bits donné et utiliser le débit minimal pour représenter ces combinaisons. L'indice peut alors être représenté par un codage de type « code de la permutation » + « offset de la combinaison » ; par exemple dans l'exemple où on code sur un indice de 4 bits les 16 combinaisons de débit comprenant

4 permutations de (13.2, 13.2, 13.2, 9.6) et 12 permutations de (16.4, 13.2, 9.6, 9.6), on pourra utiliser les indices 0-3 pour coder les 4 premières permutations possibles (avec un offset à 0 et un code allant de 0 à 3) et les indices 4-15 pour coder les 12 autres permutations possibles (avec un offset à 4 et un code de 0 à 11).

**[0097]** En référence à nouveau à la figure 3, le bloc de multiplexage 350 prend en entrée les  $n$  canaux matricés venant du bloc 310 et les débits alloués à chaque canal venant du bloc 320 pour coder ensuite séparément les différents canaux avec un codec coeur qui correspond au codec EVS par exemple. Si le codec coeur utilisé permet un codage stéréo ou multicanal, l'approche multi-mono peut être remplacée par un codage multi-stéréo ou multicanal. Une fois les canaux codés, le train binaire associé est envoyé au multiplexeur (bloc 350).

Dans les trames où une partie du budget global n'est pas utilisé en totalité, le multiplexeur (bloc 350) peut ajouter des

bits de bourrage à zéro pour atteindre le budget de bits alloué à la trame courante, soit  $B - \sum_{i=1}^n b_{t,i}^{opt}$  bits. Dans des variantes, le budget de bits restant peut être redistribué au codage des canaux transformés afin d'utiliser tout le budget disponible et si le codage multimono est basé sur une technologie de type EVS, on peut alors modifier l'algorithme de codage 3GPP EVS spécifié pour introduire des débits supplémentaires. Dans ce cas, il est également possible d'intégrer ces débits supplémentaires dans la table définissant la correspondance entre  $b_i$  et  $Q(b_i)$ .

On peut en outre réserver un bit pour pouvoir commuter entre deux modes de codage :

- Codage selon l'invention avec codage de la matrice de rotation, et
- Codage selon l'invention avec une matrice de rotation restreinte à la matrice identité (donc non transmise) ce qui revient à un codage multi-mono direct si la matrice de rotation de la trame précédente était aussi une matrice identité (par exemple quand le signal ambisonique comprend des sources sonores très diffuses ou de multiples sources étalées spatialement autour de certaines directions privilégiées, auquel cas les canaux ambisoniques sont moins corrélés que pour des sons mélangeant des sources plus ponctuelles et isolées).

Le choix entre ces deux modes implique d'utiliser un bit dans le train pour indiquer si la trame courante utilise une matrice de rotation restreinte à la matrice identité sans transmission de paramètres de rotation (bit=0) ou si une matrice de rotation est codée (bit=1). Quand bit=0, on pourra dans des variantes utiliser une allocation des bits fixes aux canaux séparés et ne pas transmettre d'allocation binaire.

**[0098]** On se réfère maintenant à la figure 4 pour décrire en détail le bloc 310 appliquant l'analyse et la transformation PCA/KLT. Dans ce bloc, le codeur calcule la matrice de covariance à partir des canaux ambisoniques (prétraités) dans le bloc 400 :

$$C = \frac{1}{L-1} \mathbf{X}\mathbf{X}^T$$

**[0099]** En variante, cette matrice peut être remplacée par la matrice de corrélation, où les canaux sont pré-normalisés par leur écart-type respectif, ou de façon générale des pondérations reflétant une importance relative peuvent être appliquées à chacun des canaux ; de plus le terme de normalisation  $1/(L-1)$  peut être omis ou remplacé par une autre valeur (par exemple  $1/L$ ). Les valeurs  $C_{ij}$  correspondent à la variance entre  $x_i$  et  $x_j$ .

**[0100]** Le codeur effectue ensuite dans le bloc 410 une décomposition en valeurs propres (EVD pour « Eigenvalue Décomposition » en anglais), en calculant les valeurs propres et les vecteurs propres de la matrice  $C$ . Les vecteurs propres sont notés ici  $\mathbf{V}_t$  pour indiquer l'indice de trame  $t$  car les vecteurs propres  $\mathbf{V}_{t-1}$  obtenus dans la trame précédente d'indice  $t-1$  sont préférentiellement mémorisés et utilisés par la suite. Les valeurs propres sont notées  $\lambda_1, \lambda_2, \dots, \lambda_n$ .

**[0101]** Dans une variante, une décomposition en valeurs singulière (SVD) des canaux prétraités  $\mathbf{X}$  peut être utilisée. On obtient ainsi les vecteurs singuliers (à gauche  $\mathbf{U}$  et droite  $\mathbf{V}$ ) et les valeurs singulières  $\sigma_i$ . Dans ce cas on peut

considérer que les valeurs propres  $\lambda_i$  sont  $\lambda_i = \sigma_i^2$  et les vecteurs propres  $\mathbf{V}_t$  sont donnés par les  $n$  vecteurs (colonne) singuliers à gauche  $\mathbf{U}$ .

**[0102]** Le codeur applique ensuite dans le bloc 420 une première permutation signée des colonnes de la matrice de transformation pour la trame  $t$  (dont les colonnes sont les vecteurs propres) afin d'éviter trop de disparité avec la matrice de transformation de la trame précédente  $t-1$ , ce qui engendrerait des problèmes de clics à la frontière avec la trame précédente.

Ainsi, une fois qu'un ébauche de la matrice de transformation est obtenue pour la trame  $t$ , le bloc 430 prend  $n$  vecteurs propres estimés  $\mathbf{V}_t = \mathbf{v}_{t,0}, \dots, \mathbf{v}_{t,n}$  de la trame courante d'indice  $t$  et  $n$  vecteurs propres  $\mathbf{V}_{t-1}$  mémorisés de la trame précédente d'indice  $t-1$ , et applique une permutation signée sur les vecteurs estimés  $\mathbf{V}_t$  pour qu'ils soient le plus proche

possible de  $V_{t-1}$ . Ainsi les vecteurs propres de la trame  $t$  sont permutés pour que la base associée soient la plus proches possibles de la base de la trame  $t - 1$ . Cela a pour effet d'améliorer la continuité des trames de signaux transformés (une fois la matrice de transformation appliquée aux canaux).

**[0103]** Une autre contrainte est que la matrice de transformation doit correspondre à une rotation. Cette contrainte permet de garantir que le codeur puisse convertir la matrice de transformation en des angles d'Euler généralisés (bloc 430) pour les quantifier (bloc 440) avec un budget de bits prédéterminé comme vu précédemment. A cet effet, le déterminant de cette matrice doit être positif (égal à +1 typiquement).

**[0104]** Prêférentiellement, la permutation signée optimale est obtenue en deux étapes :

- La première étape (S4 sur la figure 2 précédemment présentée) fait correspondre les vecteurs les plus proches entre deux trames en se souciant uniquement de l'axe et non de la direction (du sens) de l'axe. Ce problème peut être formulé comme un problème combinatoire d'affectation de tâches, où l'objectif est de trouver la configuration qui minimise un coût. Le coût peut être défini ici comme la trace de la valeur absolue de l'inter-corrélation entre les matrices de vecteurs propres des trames  $t$  et  $t - 1$ .

$$C_t = \text{tr}(\text{abs}(\text{corr}(V_t, V_{t-1})))$$

où  $\text{tr}(\cdot)$  désigne la trace d'une matrice,  $\text{abs}(\cdot)$  revient à appliquer l'opération de valeur absolue à tous les coefficients d'une matrice et  $\text{corr}(V1, V2)$  donne la matrice de corrélation entre les vecteurs  $V1$  et  $V2$ .

Dans un mode de réalisation la méthode « hongroise » (ou « algorithme hongrois ») sert à déterminer l'assignation optimale qui donne une permutation des vecteurs propres de la trame  $t$ ;

- La seconde étape (S6 sur la figure 2) consiste à déterminer la direction/sens de chaque vecteur propre permuté. Le bloc 420 calcule l'inter-corrélation entre les vecteurs propres permutés  $\tilde{V}_t$  de la trame  $t$  et le vecteur propre de la trame  $t - 1$

$$\Gamma_t = \text{corr}(\tilde{V}_t, V_{t-1})$$

**[0105]** Si une valeur sur la diagonale de la matrice d'inter-corrélation  $\Gamma_t$  est négative, cela dénote un changement de signe entre les directions de vecteurs propres. Une inversion de signe est alors opérée sur le vecteur propre correspondant dans  $\tilde{V}_t$

A l'issue des deux étapes la matrice de transformation à la trame  $t$  est désignée par  $V_t$  de sorte qu'à la trame suivante la matrice mémorisée devienne  $V_{t+1}$ .

**[0106]** Dans une variante, la recherche de la permutation signée optimale peut se faire en calculant la matrice de

passage  $V_{t-1}^{-1}V_t$  ou  $V_t V_{t-1}^{-1}$  qui est convertie en 3D ou 4D et en convertissant cette matrice de passage respectivement en un quaternion unitaire ou deux quaternions unitaires. La recherche devient alors une recherche du plus proche voisin avec un dictionnaire représentant l'ensemble des permutations signées possibles. Par exemple dans le cas 4D les douze permutations paires possibles (sur 24 permutations totales) de 4 valeurs sont associées aux doubles quaternions unitaires suivants écrits comme des vecteurs 4D:

- (1, 0, 0, 0) et (1, 0, 0, 0)
- (0, 0, 0, 1) et (0, 0, -1, 0)
- (0, 1, 0, 0) et (0, 0, 0, -1)
- (0, 0, 1, 0) et (0, -1, 0, 0)]
- (0.5, -0.5, -0.5, -0.5) et (0.5, 0.5, 0.5, 0.5)
- (0.5, 0.5, 0.5, 0.5) et (0.5, -0.5, -0.5, -0.5)
- (0.5, -0.5, 0.5, -0.5) et (0.5, -0.5, 0.5, 0.5)
- (0.5, -0.5, 0.5, 0.5) et (0.5, -0.5, -0.5, 0.5)
- (0.5, 0.5, -0.5, 0.5) et (0.5, 0.5, -0.5, -0.5)
- (0.5, -0.5, -0.5, 0.5) et (0.5, 0.5, -0.5, 0.5)
- (0.5, 0.5, -0.5, -0.5) et (0.5, 0.5, 0.5, -0.5)
- (0.5, 0.5, 0.5, -0.5) et (0.5, -0.5, 0.5, -0.5)

La recherche de la permutation (paire) optimale peut se faire en utilisant la liste ci-dessus comme un dictionnaire de double quaternion prédéfini et en effectuant une recherche du plus proche voisin par rapport au double quaternion

associé à la matrice de passage. Un avantage de cette méthode est de réutiliser les paramètres de rotation de type quaternion et double quaternion.

**[0107]** L'opération qui est mise en oeuvre dans le bloc suivant 460 suppose que la matrice de transformation après permutation signée est bien une matrice de rotation ; la matrice de transformation est forcément unitaire, mais il faut également que son déterminant soit égal à 1

$$\det(V_t) = 1$$

**[0108]** Or la matrice de transformation issue des blocs 410 et 420 (après EVD et permutations signées) est une matrice orthogonale (unitaire) pouvant avoir un déterminant à -1 ou 1, c'est-à-dire une matrice de réflexion ou de rotation.

**[0109]** Si la matrice de transformation est une matrice de réflexion (si son déterminant est égal à -1), elle peut être modifiée en une matrice de rotation en inversant un vecteur propre (par exemple le vecteur propre associé à la plus faible valeur) ou en intervertissant deux colonnes (vecteurs propres).

Certaines méthodes de décomposition en valeurs propres (par exemple par rotation de Givens) ou de décomposition en valeurs singulières peuvent conduire à des matrices de transformation qui sont intrinsèquement des matrices de rotation (avec un déterminant à +1) ; dans ce cas, l'étape de vérification que le déterminant est +1 sera optionnelle.

**[0110]** Le bloc 430 convertit la matrice de rotation en paramètres. Dans le mode de réalisation privilégié, on utilise une représentation angulaire pour la quantification (6 angles d'Euler généralisés pour le cas 4D, 3 angles d'Euler pour le cas 3D, et un angle en 2D). Pour le cas ambisonique (quatre canaux) on obtient six angles d'Euler généralisés selon la méthode décrite dans l'article « Generalization of Euler Angles to N-Dimensional Orthogonal Matrices » de David K. Hoffman, Richard C. Raffenetti, and Klaus Ruedenberg, paru dans Journal of Mathematical Physics 13, 528 (1972); pour le cas de l'ambisonique planaire (trois canaux) on obtient trois angles d'Euler et pour le cas stéréo on obtient un angle de rotation selon les méthodes bien connues de l'état de l'art. Les valeurs des angles sont quantifiées dans le bloc 440 avec budget prédéterminé de bits. Dans le mode de réalisation privilégié une quantification scalaire est utilisée et le pas de quantification est par exemple identique pour chaque angle. Par exemple dans le cas de 4 canaux, on code 6 angles d'Euler généralisés avec  $3 \times (8+9) = 51$  bits (3 angles définis sur un intervalle de  $[-\pi/2, \pi/2]$  codés sur 8 bits avec un pas de  $\pi/256$  et les 3 autres angles définis sur un intervalle de  $[-\pi, \pi]$  codés sur 9 bits avec un pas de  $\pi/256$ ). Les indices de quantification de la matrice de transformation sont envoyés au multiplexeur (bloc 350). De plus, le bloc 440 pourra convertir les paramètres quantifiés en une matrice de rotation quantifiée  $\hat{V}_t$  si les paramètres utilisés pour la quantification ne correspondent pas aux paramètres utilisés pour l'interpolation.

**[0111]** En variante, les blocs 430 et 440 peuvent être remplacés comme suit :

- Le bloc 430 peut effectuer une conversion des matrices de rotations en un double quaternion unitaire (cas de 4 canaux), en quaternion unitaire (cas de 3 canaux) et en un angle (cas de 2 canaux).

**[0112]** Cette conversion en double quaternion pour le cas 4D pourra être réalisée pour une matrice de rotation dont les coefficients sont notés  $R[i,j]$ ,  $i,j=0\dots3$ , par le pseudo-code suivante :

Calcul de la matrice associée  $A[i,j]$  avec :

$$A[0,0] = R[0,0] + R[1,1] + R[2,2] + R[3,3]$$

$$A[1,0] = R[1,0] - R[0,1] + R[3,2] - R[2,3]$$

$$A[2,0] = R[2,0] - R[3,1] - R[0,2] + R[1,3]$$

$$A[3,0] = R[3,0] + R[2,1] - R[1,2] - R[0,3]$$

$$A[0,1] = R[1,0] - R[0,1] - R[3,2] + R[2,3]$$

$$A[1,1] = -R[0,0] - R[1,1] + R[2,2] + R[3,3]$$

EP 3 706 119 A1

$$A[2,1] = -R[3,0] - R[2,1] - R[1,2] - R[0,3]$$

5  $A[3,1] = R[2,0] - R[3,1] + R[0,2] - R[1,3]$

$$A[0,2] = R[2,0] + R[3,1] - R[0,2] - R[1,3]$$

10  $A[1,2] = R[3,0] - R[2,1] - R[1,2] + R[0,3]$

$$A[2,2] = -R[0,0] + R[1,1] - R[2,2] + R[3,3]$$

15  $A[3,2] = -R[1,0] - R[0,1] - R[3,2] - R[2,3]$

20  $A[0,3] = R[3,0] - R[2,1] + R[1,2] - R[0,3]$

$$A[1,3] = -R[2,0] - R[3,1] - R[0,2] - R[1,3]$$

25  $A[2,3] = R[1,0] + R[0,1] - R[3,2] - R[2,3]$

$$A[3,3] = -R[0,0] + R[1,1] + R[2,2] - R[3,3]$$

30

$$A=A/4$$

Calcul des 2 quaternions à partir de la matrice associée

35

A2 = square(A) # carré de coefficients  
q1 = sqrt(A2.sum(axis=1)) # somme sur le lignes  
q2= sqrt(A2.sum(axis=0)) # somme sur les colonnes

40 Détermination des signes

Pour k=0..3 : Si sign(A[i,k])<0, Alors q2[k] = -q2[k]  
Pour k=0..3 : Si sign(A[k,j])!=sign(q1[k]\*q2[j]), Alors q1[k] = -q1[k]

45 La conversion en quaternion pour le cas 3D peut être réalisée comme suit pour une matrice R[i,j] i,j=0...2 de taille 3x3:  
Calcul de la matrice associée simplifiée :

50  $q[0]=(R[0,0]+R[1,1]+R[2,2]+1)^2 + (R[2,1]-R[1,2])^2 + (R[0,2]-R[2,0])^2 + (R[1,0]-R[0,1])^2$

55  $q[1]=(R[2,1]-R[1,2])^2 + (R[0,0]-R[1,1]-R[2,2]+1)^2 + (R[1,0]+R[0,1])^2 + (R[2,0]+R[0,2])^2$

$$q[2]=(R[0,2]-R[2,0])^2 + (R[1,0]+R[0,1])^2 + (R[1,1]-R[0,0]-R[2,2]+1)^2 + (R[2,1]+R[1,2])^2$$

5

$$q[3]=(R[1,0]-R[0,1])^2 + (R[2,0]+R[0,2])^2 + (R[2,1]+R[1,2])^2 + (R[2,2]-R[0,0]-R[1,1]+1)^2$$

10

Pour  $i=0..3$  :  $q[i] = \text{sqrt}(q[i])/4$

**[0113]** Calcul du quaternion  $q$

Si

15

$$(R[2,1]-R[1,2])<0, q[1]=-q[1]$$

Si

20

$$(R[0,2]-R[2,0])<0, q[2]=-q[2]$$

Si

25

$$(R[1,0]-R[0,1])<0, q[3]=-q[3]$$

**[0114]** Le calcul de l'angle pour le cas d'une matrice 2x2 se fait selon les méthodes de l'état de l'art déjà connue.

30 Dans des variantes on pourra convertir les quaternions unitaires  $q_1$ ,  $q_2$  (cas 4D) et  $q$  (cas 3D) en des représentations axe-angle connues de l'état de l'art.

- Le bloc 440 peut réaliser une quantification dans le domaine indiqué :

35

\* Cas de 4 canaux : la paire de quaternions unitaires  $q_1$  et  $q_2$  est quantifiée par un dictionnaire de quantification sphérique en dimension 4 ; par convention on quantifie  $q_1$  avec un dictionnaire hémisphérique (car  $q_1$  et  $-q_1$  correspondent à une même rotation 3D) et  $q_2$  est quantifié avec un dictionnaire sphérique. Des exemples de dictionnaires peuvent être donnés par des points prédéfinis à partir de polyèdres de dimension 4 ; dans des variantes on pourra quantifier une double représentation axe-angle associés qui serait équivalente au double quaternion ;

40

\* Cas de 3 canaux : le quaternion unitaire est quantifié par un dictionnaire de quantification sphérique en dimension 4 - des exemples de dictionnaires peuvent être donnés par des points prédéfinis à partir de polyèdres de dimension 4 ;

\* Cas de 2 canaux : l'angle est quantifié par quantification scalaire uniforme.

45

**[0115]** On décrit maintenant le bloc 460 d'interpolation des matrices de rotation entre deux trames successives. Il permet de lisser les discontinuités des canaux après l'application de ces matrices. Typiquement, si deux jeux d'angles ou de quaternions sont trop différents d'une trame précédente  $t-1$  à la suivante  $t$ , des clics audibles sont à craindre s'il n'a pas été pratiqué entre ces deux trames une transition lissée dans des sous-trames entre ces deux trames. On réalise alors une interpolation de passage entre la matrice de rotation calculée pour la trame  $t-1$  et la matrice de rotation calculée pour la trame  $t$ . Le codeur interpole dans le bloc 460 la représentation (quantifiée) de la rotation entre la trame courante et de la trame précédente pour éviter des fluctuations trop rapides des différents canaux après transformation. Le nombre d'interpolations peut être fixe (égal à une valeur prédéterminée) ou adaptatif. Chaque trame est alors divisée en sous-trames en fonction du nombre d'interpolations déterminé dans le bloc 450. Ainsi, si une interpolation adaptative est utilisée, le bloc 450 peut coder sur un nombre de bits choisi le nombre d'interpolations à effectuer, et donc le nombre de sous-trames à prévoir, dans le cas où ce nombre est déterminé de façon adaptative ; dans le cas d'une interpolation fixe, aucune information n'est à coder.

55

**[0116]** Ensuite, le bloc 460 convertit les matrices de rotation dans un domaine spécifique représentant une matrice

de rotations. La trame est découpée en sous-trames, et dans le domaine choisi l'interpolation est effectuée pour chaque sous-trame.

Pour un signal d'entrée ambisonique d'ordre 1 (à 4 canaux W, X, Y, Z), dans le bloc 460, le codeur reconstruit à partir des 6 angles d'Euler quantifiés une matrice de rotation 4D quantifiée et celle-ci est ensuite convertie en deux quaternions unitaires à des fins d'interpolation. Dans une variante où l'entrée du codeur est un signal ambisonique planaire (3 canaux W, X, Y), dans le bloc 460 le codeur reconstruit à partir des 3 angles d'Euler quantifiés une matrice de rotation 3D quantifiée et celle-ci est ensuite convertie en un quaternion unitaire à des fins d'interpolation. Dans une variante où l'entrée du codeur est un signal stéréo, le codeur utilise dans le bloc 460 la représentation de la rotation 2D quantifiée avec un angle de rotation.

Dans le mode de réalisation avec 4 canaux, pour l'interpolation de la matrice de rotation entre la trame t et la trame t-1, la matrice de rotation calculée pour la trame t est factorisée en 2 quaternions (un double quaternion) grâce à la factorisation de Cayley et on utilise le double quaternion mémorisé pour la trame précédente t-1 et noté (Q<sub>L,t-1</sub>, Q<sub>R,t-1</sub>). **[0117]** Pour chaque sous-trame, on interpole dans chaque sous-trame les quaternions deux à deux.

Pour le quaternion gauche (Q<sub>L,t</sub>), le bloc détermine le plus court chemin entre les deux possible (Q<sub>L,t</sub> ou -Q<sub>L,t</sub>). Selon les cas, on inverse le signe du quaternion de la trame courante. Puis l'interpolation est calculée pour le quaternion gauche avec l'interpolation sphérique linéaire (SLERP) :

$$Q_{L,interp}(\alpha) = Q_{L,t-1} \frac{\sin(1-\alpha)\Omega_L}{\sin \Omega_L} + Q_{L,t} \frac{\sin \alpha\Omega_L}{\sin \Omega_L}$$

où α correspond au facteur d'interpolation (α=1/K, 2/K, ... 1) et Ω<sub>L</sub> = arccos(Q<sub>L,t-1</sub> · Q<sub>L,t</sub>)

**[0118]** Pour le quaternion droit (Q<sub>R,t</sub>), s'il y a eu une inversion pour le quaternion gauche alors il faut respecter la parité et forcer le signe du quaternion droit. Cette contrainte de signe est appelée ci-après « contrainte de plus court chemin conjoint ». Puis l'interpolation est calculée de manière similaire au quaternion gauche :

$$Q_{R,interp}(\alpha) = Q_{R,t-1} \frac{\sin(1-\alpha)\Omega_R}{\sin \Omega_R} + Q_{R,t} \frac{\sin \alpha\Omega_R}{\sin \Omega_R}$$

où α correspond au facteur d'interpolation (α=1/K, 2/K, ... 1) et Ω<sub>R</sub> = arccos(Q<sub>R,t-1</sub> · Q<sub>R,t</sub>)

**[0119]** Une fois l'interpolation calculée pour les deux quaternions, on calcule la matrice de rotation de dimension 4x4 (respectivement 3x3 pour l'ambisonique planaire ou 2x2 pour le cas stéréo).

Cette conversion en matrice de rotation peut être effectuée selon les pseudo-codes suivants :

Cas 4D : pour un double quaternion

- Comme décrit précédemment on calcule les matrices de quaternion et anti-quaternion et on calcule le produit matriciel.

Cas 3D : pour quaternion q=(w,x,y,z) on obtient la matrice M[i,j], i,j=0...2, de taille 3x3

$$xy=2*x*y$$

$$xz=2*x*z$$

$$yz=2*y*z$$

$$wx=2*w*x$$

$$wy=2*w*y$$

EP 3 706 119 A1

$$wz=2*w*z$$

5

$$xx=2*x*x$$

$$yy=2*y*y$$

10

$$zz=2*z*z$$

$$M[0][0] = 1-(yy+zz)$$

15

$$M[0][1] = (xy-wz)$$

$$M[0][2] = (xz+wy)$$

20

$$M[1][0] = (xy+wz)$$

$$M[1][1] = 1-(xx+zz)$$

25

$$M[1][2] = (yz-wx)$$

30

$$M[2][0] = (xz-wy)$$

$$M[2][1] = (yz+wx)$$

35

$$M[2][2] = 1-(xx+yy);$$

40

**[0120]** Enfin, les matrices  $V_t^{interp}(\alpha)$  (ou leurs transposées) calculées par sous-trame dans le bloc 460 d'interpolation sont ensuite utilisées dans le bloc 470 de transformation qui produit n canaux transformés par application des matrices de rotation ainsi trouvées, aux canaux ambisoniques qui ont été prétraités par le bloc 300.

45

**[0121]** On revient ci-après sur le nombre K de sous-trames à déterminer dans le bloc 450 pour le cas où ce nombre est adaptatif. Il est mesuré l'écart final entre la trame courante et la trame précédente ou directement à partir de la différence angulaire des paramètres décrivant la matrice de rotation. On cherche dans ce dernier cas à faire en sorte que la variation angulaire entre sous-trames successives ne soit pas perceptible. La réalisation d'un nombre de sous-trames adaptatif est surtout avantageuse pour réduire la complexité moyenne du codec mais s'il est choisi de réduire la complexité on peut préférer utiliser une interpolation avec un nombre fixe de sous-trames.

50

**[0122]** L'écart final entre la matrice de rotation corrigée de la trame t et la matrice de rotation de la trame t - 1 donne une mesure de l'importance de la différence de matricage des canaux entre les deux trames. Plus cet écart est important et plus le nombre de sous-trames pour l'interpolation faite dans le bloc 460 est élevé. On utilise la somme de la valeur absolue de la matrice d'inter-corrélation entre la matrice de transformation de la trame courante et la trame précédente, comme suit, pour mesurer cet écart :

55

$$\delta_t = \|I_n - \mathit{corr}(V_t, V_{t-1})\|$$

où  $I_n$  est la matrice identité,  $V_t$  les vecteurs propres à la trame d'indice  $t$ , et  $\|M\|$  est une norme de la matrice  $M$  qui correspond ici à la somme des valeurs absolues de tous les coefficients. D'autres normes matricielles peuvent être utilisées (par exemple la norme de Frobenius).

**[0123]** Dans le cas où les deux matrices sont identiques alors cet écart est égal à 0. Plus les matrices sont dissimilaires, plus la valeur de l'écart  $\delta_t$  est élevée. Des seuils prédéterminés peuvent être appliqués à  $\delta_t$ , à chaque seuil est associé un nombre prédéfini d'interpolations par exemple selon la logique de décision suivante :

Seuils : {4.0, 5.0, 6.0, 7.0}

Nombre  $K$  des sous-trames pour interpolation : {10, 48, 96, 192}

Ainsi seuls deux bits peuvent suffire à coder les quatre valeurs possibles donnant le nombre de subdivisions (sous-trames).

Le nombre  $K$  d'interpolations déterminé par le bloc 450 est ensuite envoyé au module d'interpolation 460 et dans le cas adaptatif le nombre de sous-trames est codé sous la forme d'un indice binaire qui est envoyé au multiplexeur (bloc 350).

**[0124]** La réalisation de l'interpolation permet d'appliquer *in fine* une optimisation de la décorrélation des canaux d'entrée avant codage multi-mono. En effet, les matrices de rotation calculées respectivement pour une trame précédente  $t-1$  et une trame courante  $t$  peuvent être très différentes du fait de cette recherche de décorrélation, mais l'interpolation permet néanmoins de lisser cette différence.

L'interpolation utilisée nécessite un coût de calcul limité pour le codeur et le décodeur puisqu'elle est réalisée dans un domaine spécifique (angle en 2D, quaternion en 3D, double quaternion en 4D). Cette approche est plus avantageuse que d'interpoler des matrices de covariance calculées pour l'analyse PCA/KLT et de répéter plusieurs fois par trame une décomposition en valeurs propres type EVD (pour « EigenValue Décomposition »).

**[0125]** Le bloc 470 effectue ensuite le matriçage des canaux ambisoniques par sous-trame à l'aide des matrices de transformation calculées dans le bloc 460. Ce matriçage revient à calculer par sous-trame  $V_t^{interp}(\alpha)^T X(\alpha)$ , où  $X(\alpha)$  correspond aux sous-blocs de taille  $n \times (L/K)$  pour  $\alpha=1/K, 2/K, \dots, 1$ . Le signal contenu dans ces canaux est ensuite envoyé au bloc 340 pour l'encodage multi-monos.

**[0126]** On se réfère maintenant à la figure 5 pour décrire un décodeur dans un exemple de réalisation de l'invention. Après démultiplexage du train binaire pour la trame courante  $t$  par le bloc 500, l'information d'allocation est décodée (bloc 510) ce qui permet de dé-multiplexer et de décoder (bloc 520) le(s) train(s) binaire(s) reçu(s) pour chacun des  $n$  canaux transformés.

Le bloc 520 fait appel à plusieurs instances exécutées séparément du décodage coeur. Le décodage coeur peut être de type EVS éventuellement modifié pour améliorer ses performances. Selon une approche multi-mono, chaque canal est décodé séparément. Si le codage précédemment utilisé est un codage stéréo ou multicanal, l'approche multi-mono peut être remplacée par un multi-stéréo ou multicanal pour le décodage. Les canaux ainsi décodés sont envoyés au bloc 530 qui décode la matrice de rotation pour la trame courante et de façon optionnelle le nombre  $K$  de sous-trames à utiliser pour l'interpolation (si l'interpolation est adaptative). Pour chaque matrice, le bloc d'interpolation 460 découpe la trame en sous-trames dont le nombre  $K$  peut être lu dans le flux codé par le bloc 610 (figure 6) et interpole les matrices de rotation, le but étant de retrouver - en l'absence d'erreurs de transmission - les mêmes matrices que dans le bloc 460 du codeur pour pouvoir inverser la transformation qui a été précédemment faite dans le bloc 470.

**[0127]** Le bloc 530 effectue le matriçage inversant celui du bloc 470 pour reconstruire un signal décodé, comme

détaillé ci-après en référence à la figure 6. Ce matriçage revient à calculer par sous-trame  $V_t^{interp}(\alpha)\hat{X}(\alpha)$ , où  $\hat{X}(\alpha)$  correspond aux sous-blocs successifs de taille  $n \times (L/K)$  pour  $\alpha=1/K, 2/K, \dots, 1$ .

Le bloc 530 effectue globalement le décodage et de la synthèse PCA/KLT inverse qui a été effectué par le bloc 310 de la figure 3. Les indices de quantification des paramètres de quantification de la rotation dans la trame courante sont décodés dans le bloc 600. Une quantification scalaire peut être utilisée et le pas de quantification est identique pour chaque angle. Dans le cas adaptatif le nombre de sous-trames d'interpolation est décodé (bloc 610) pour retrouver le nombre  $K$  de sous-trames parmi l'ensemble {10, 48, 96, 192} ; dans des variantes où la longueur de trames  $L$  est différente, cet ensemble de valeurs pourra être adapté. L'interpolation du décodeur est identique à celle effectuée à l'encodeur (bloc 460).

**[0128]** Le bloc 620 effectue le matriçage inverse des canaux ambisoniques par sous-trame à l'aide des inverses (les transposées en pratique) des matrices de transformation calculées dans le bloc 460.

**[0129]** Ainsi, l'invention utilise une toute autre approche que le codec MPEG-H à addition/recouvrement en se basant sur une représentation spécifique des matrices de transformation qui sont restreintes à des matrices de rotation d'une trame à l'autre, dans le domaine temporel, permettant notamment une interpolation des matrices de transformation,

avec une mise en correspondance qui assure une cohérence en direction (y compris en prenant en compte le sens par le signe).

**[0130]** L'approche générale de l'invention est un codage de sons ambisoniques dans le domaine temporel par PCA avec notamment des matrices de transformation PCA forcées à être des matrices de rotations et interpolées par sous-trames de façon optimisée (en particulier dans le domaine des quaternions/doubles quaternions) pour améliorer la qualité. Le pas d'interpolation est soit fixe, soit adaptatif en fonction d'un critère d'écart entre une matrice d'inter-corrélation et une matrice de référence (identité) ou entre matrices à interpoler. La quantification des matrices de rotation peut être mise en oeuvre dans le domaine des angles d'Euler généralisés. Cependant il peut être choisi préférentiellement de quantifier les matrices de dimension 3 et 4 dans le domaine des quaternions et doubles quaternions (respectivement), ce qui permet de rester dans le même domaine pour la quantification et l'interpolation.

En outre, un alignement des vecteurs propres est utilisé pour éviter les problèmes de clics et d'inversion de canaux, d'une trame à l'autre.

**[0131]** Bien entendu, la présente invention ne se limite pas aux formes de réalisation décrites ci-avant à titre d'exemple et s'étend à d'autres variantes.

**[0132]** Ainsi, la description précédente a traité les cas de quatre canaux.

Néanmoins, dans des variantes, on peut également coder un nombre de canaux supérieur à quatre. La mise en oeuvre reste identique (en termes de blocs fonctionnels) au cas n=4, mais l'interpolation par double quaternion est remplacée par la méthode générale ci-après.

Les matrices de transformation aux trames  $t-1$  et  $t$  sont notées  $V_{t-1}$  et  $V_t$ . L'interpolation peut être effectuée avec un facteur  $\alpha$  entre  $V_{t-1}$  et  $V_t$  tel que :

$$V_t^{interp}(\alpha) = V_{t-1}(V_{t-1}^T V_t)^\alpha$$

**[0133]** Le terme  $(V_{t-1}^T V_t)^\alpha$  peut se calculer directement par décomposition en valeurs propres de  $V_{t-1}^T V_t$ . En

effet, si  $V_{t-1}^T V_t = QLQ^T$ , on a:  $(V_{t-1}^T V_t)^\alpha = QL^\alpha Q^T$ .

On notera également que cette variante pourrait aussi remplacer l'interpolation par double quaternion unitaire (cas 4D), quaternion unitaire (cas 3D) ou angle, cependant elle serait moins avantageuse car elle nécessiterait une étape de diagonalisation supplémentaire et des calculs de puissance, alors que le mode de réalisation décrit précédemment est plus efficace pour ces cas de 2, 3 ou 4 canaux.

## Revendications

1. Procédé de codage en compression de signaux sonores formant une succession dans le temps de trames (t-1, t) d'échantillons, dans chacun de N canaux en représentation ambisonique d'ordre supérieur à 0, le procédé comportant :

- former, à partir des canaux pour une trame courante (t), une matrice de covariance entre canaux et rechercher des vecteurs propres de la matrice de covariance pour obtenir une matrice de vecteurs propres,
- tester la matrice de vecteurs propres pour vérifier qu'elle représente une rotation dans un espace de dimension N et corriger sinon la matrice de vecteurs propres jusqu'à obtenir une matrice de rotation, pour la trame courante (t), et
- appliquer ladite matrice de rotation aux signaux des N canaux avant un encodage par canaux séparés desdits signaux.

2. Procédé selon la revendication 1, comportant en outre :

- comparer la matrice de vecteurs propres obtenue pour la trame courante (t) à une matrice de rotation obtenue pour une trame (t-1) précédant la trame courante (t), et
- permuter des colonnes de la matrice de vecteurs propres de la trame courante (t) pour assurer une cohérence avec la matrice de rotation de la trame précédente (t-1).

3. Procédé selon la revendication 2, dans lequel ladite permutation des colonnes permet d'assurer une cohérence d'axes des vecteurs, et le procédé comporte en outre :

## EP 3 706 119 A1

- vérifier, pour chaque vecteur propre de la trame courante (t), une cohérence de direction avec un vecteur-colonne de position correspondante de la matrice de rotation de la trame précédente (t-1), et
- en cas d'incohérence, inverser le signe des éléments de ce vecteur propre dans la matrice de vecteurs propres de la trame courante (t).

5

4. Procédé selon l'une des revendications précédentes, comportant en outre :

- une estimation d'écart entre la matrice de rotation obtenue pour la trame courante (t) et une matrice de rotation obtenue pour une trame (t-1) précédant la trame courante,
- en fonction de l'écart estimé, déterminer si au moins une interpolation est à opérer entre la matrice de rotation de la trame courante (t) et la matrice de rotation de la trame précédente (t-1).

10

5. Procédé selon la revendication 4, dans lequel :

- en fonction de l'écart estimé, il est déterminé un nombre d'interpolations à opérer entre la matrice de rotation de la trame courante (t) et la matrice de rotation de la trame précédente (t-1),
- la trame courante est découpée en un nombre de sous-trames correspondant au nombre d'interpolations à opérer, et
- au moins ce nombre d'interpolations est codé en vue d'une transmission via un réseau.

15

20

6. Procédé selon l'une des revendications précédentes, dans lequel une permutation entre colonnes de la matrice de vecteurs propres inversant le signe d'un déterminant de la matrice de vecteurs propres et le déterminant d'une matrice de rotation étant égal à 1,

si le déterminant de la matrice de vecteurs propres est égal à -1, on inverse les signes des éléments d'une colonne choisie de la matrice de vecteurs propres, pour que le déterminant soit égal à 1 et former ainsi une matrice de rotation.

25

7. Procédé selon l'une des revendications précédentes, dans lequel la représentation ambisonique est d'ordre 1 et le nombre N de canaux est quatre, et dans lequel la matrice de rotation de la trame courante est représentée par deux quaternions.

30

8. Procédé selon la revendication 7, prise en combinaison avec la revendication 6, dans lequel chaque interpolation pour une sous-trame courante est une interpolation sphérique linéaire (SLERP), menée en fonction de l'interpolation de la sous-trame précédant la sous-trame courante et à partir des quaternions de la sous-trame précédente.

35

9. Procédé selon la revendication 8, dans lequel l'interpolation sphérique linéaire de la sous-trame courante est menée pour obtenir les quaternions de la sous-trame courante comme suit :

$$Q_{L,interp}(\alpha) = Q_{L,t-1} \frac{\sin(1-\alpha)\Omega_L}{\sin \Omega_L} + Q_{L,t} \frac{\sin \alpha\Omega_L}{\sin \Omega_L}$$

40

$$Q_{R,interp}(\alpha) = Q_{R,t-1} \frac{\sin(1-\alpha)\Omega_R}{\sin \Omega_R} + Q_{R,t} \frac{\sin \alpha\Omega_R}{\sin \Omega_R}$$

45

Où:

$Q_{L,t-1}$  est l'un des quaternions de la sous-trame précédente t-1,  
 $Q_{R,t-1}$  est l'autre des quaternions de la sous-trame précédente t-1,  
 $Q_{L,t}$  est l'un des quaternions de la sous-trame courante t,  
 $Q_{R,t}$  est l'autre des quaternions de la sous-trame courante t,

50

$$\Omega_L = \text{Arccos} (Q_{L,t-1} \cdot Q_{L,t}) ; \Omega_R = \text{Arccos} (Q_{R,t-1} \cdot Q_{R,t})$$

55

et  $\alpha$  correspond à un facteur d'interpolation.

10. Procédé selon l'une des revendications précédentes, dans lequel la recherche des vecteurs propres est effectuée

## EP 3 706 119 A1

par analyse en composantes principales (PCA), ou par transformée de Karhunen Loeve (KLT), dans le domaine temporel.

- 5
11. Procédé selon l'une des revendications précédentes, dans lequel une étape préalable de prévision de budget d'allocation de bits par canal ambisonique est mise en oeuvre et comporte :
- pour chaque canal ambisonique, une estimation d'énergie acoustique courante dans le canal,
  - la sélection dans une mémoire d'un score prédéterminé, de qualité (MOS), fonction de ce canal ambisonique et d'un débit courant dans le réseau,
  - 10 - l'estimation d'une pondération à opérer pour l'allocation de bits à ce canal, par multiplication du score sélectionné à l'énergie estimée.
12. Procédé de décodage de signaux sonores formant une succession dans le temps de trames ( $t-1$ ,  $t$ ) d'échantillons, dans chacun de  $N$  canaux en représentation ambisonique d'ordre supérieur à 0, le procédé comportant:
- 15 - recevoir, pour une trame courante ( $t$ ), en plus des signaux des  $N$  canaux de cette trame courante, des paramètres d'une matrice de rotation,
  - construire une matrice de rotation inverse à partir desdits paramètres,
  - appliquer ladite matrice de rotation inverse à des signaux issus des  $N$  canaux reçus, avant un décodage par canaux séparés desdits signaux.
- 20
13. Dispositif de codage comportant un circuit de traitement pour la mise en oeuvre du procédé selon l'une des revendications 1 à 11.
- 25
14. Dispositif de décodage comportant un circuit de traitement pour la mise en oeuvre du procédé selon la revendication 12.
- 30
15. Programme informatique comportant des instructions pour la mise en oeuvre du procédé selon l'une des revendications 1 à 12, lorsque lesdites instructions sont exécutées par un processeur d'un circuit de traitement.
- 35
- 40
- 45
- 50
- 55

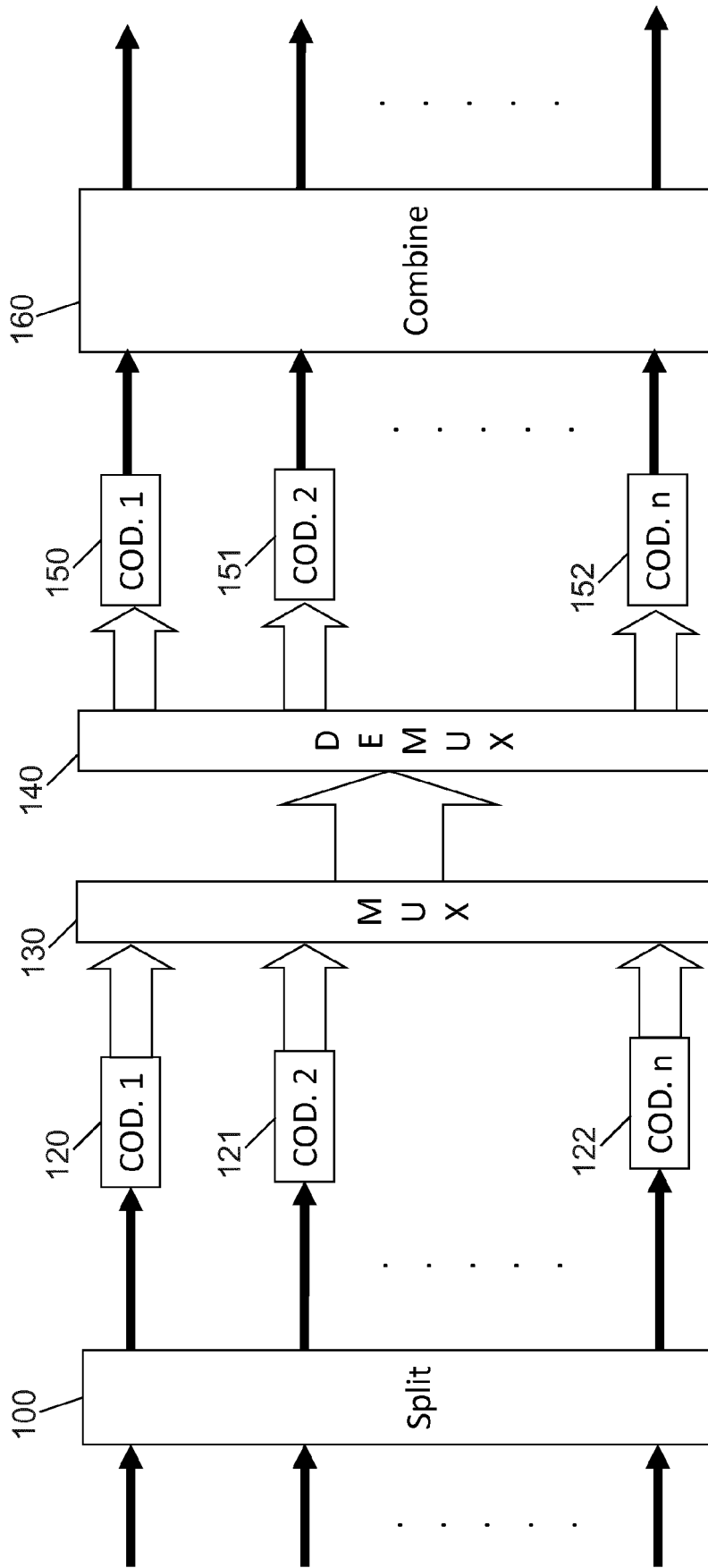


FIG. 1

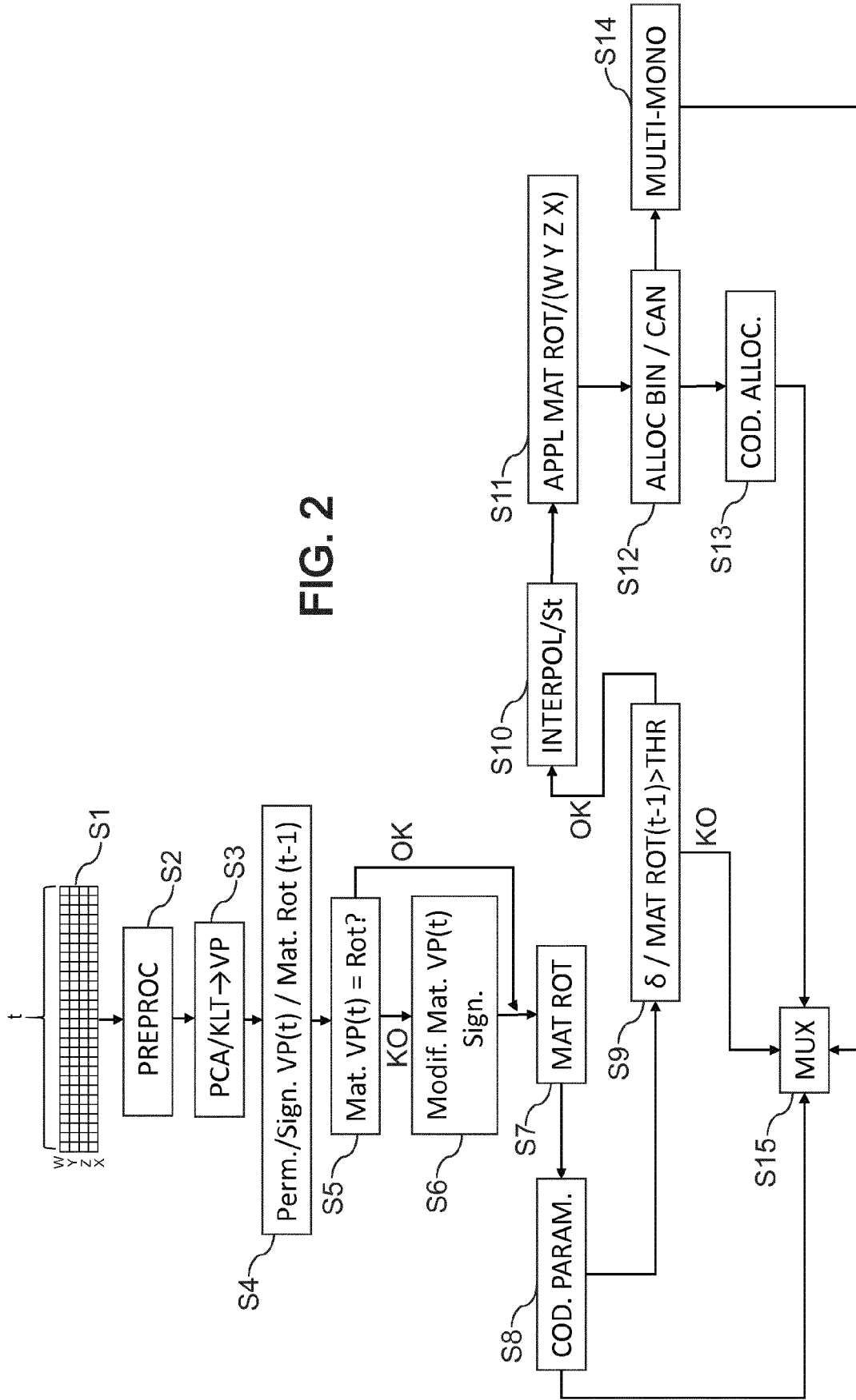


FIG. 2

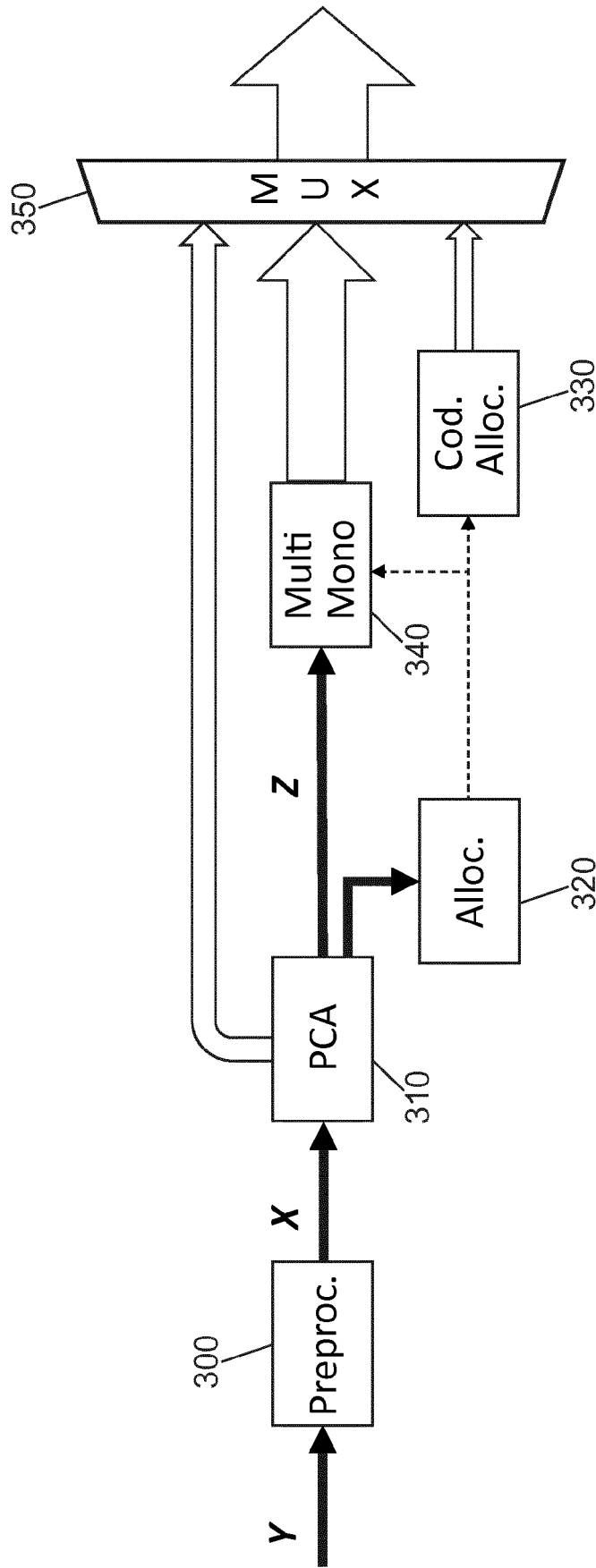


FIG. 3

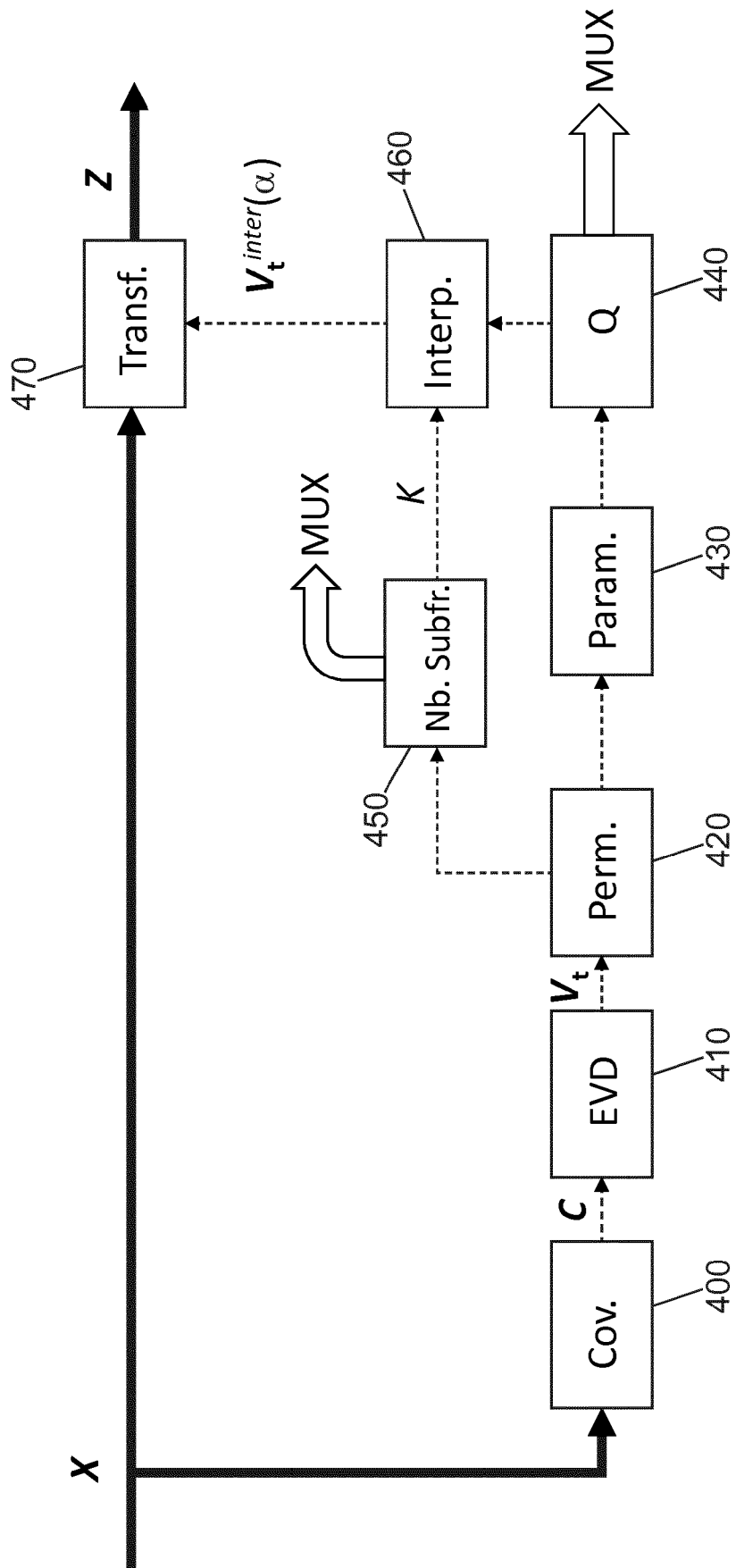


FIG. 4

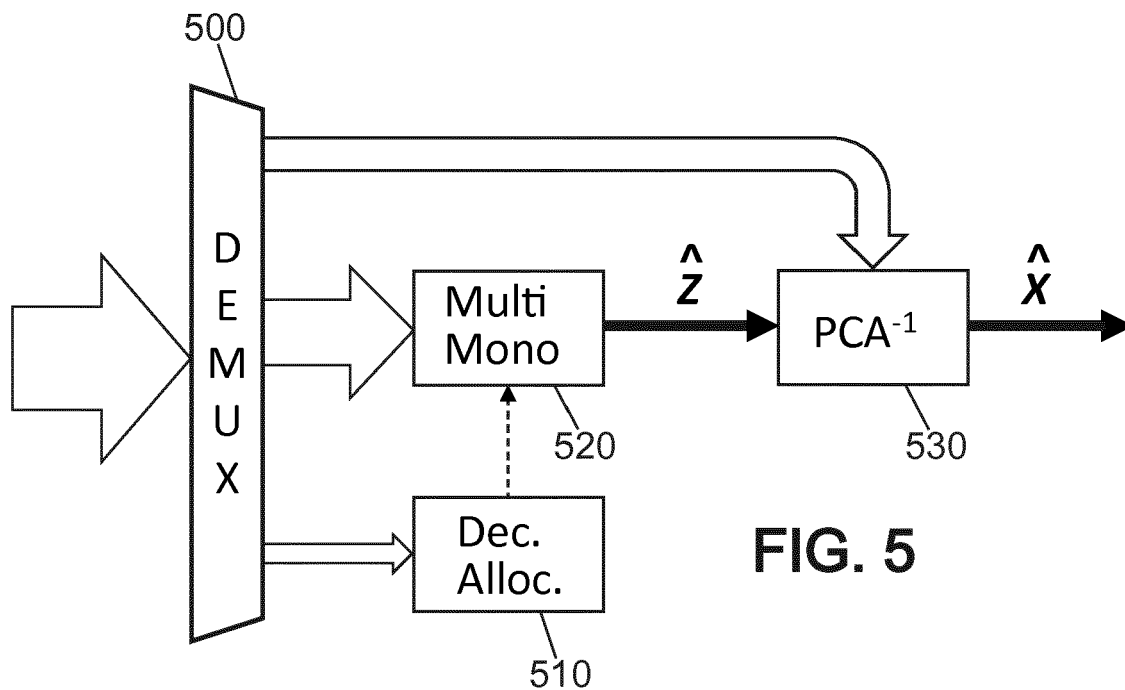


FIG. 5

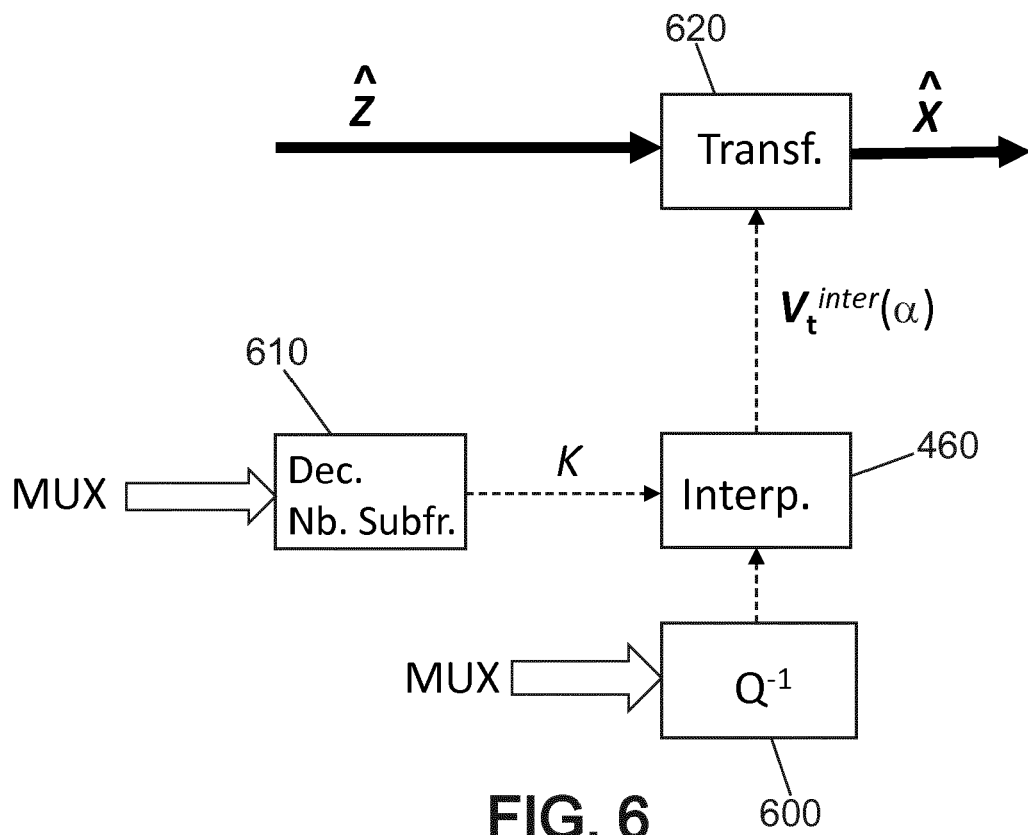


FIG. 6

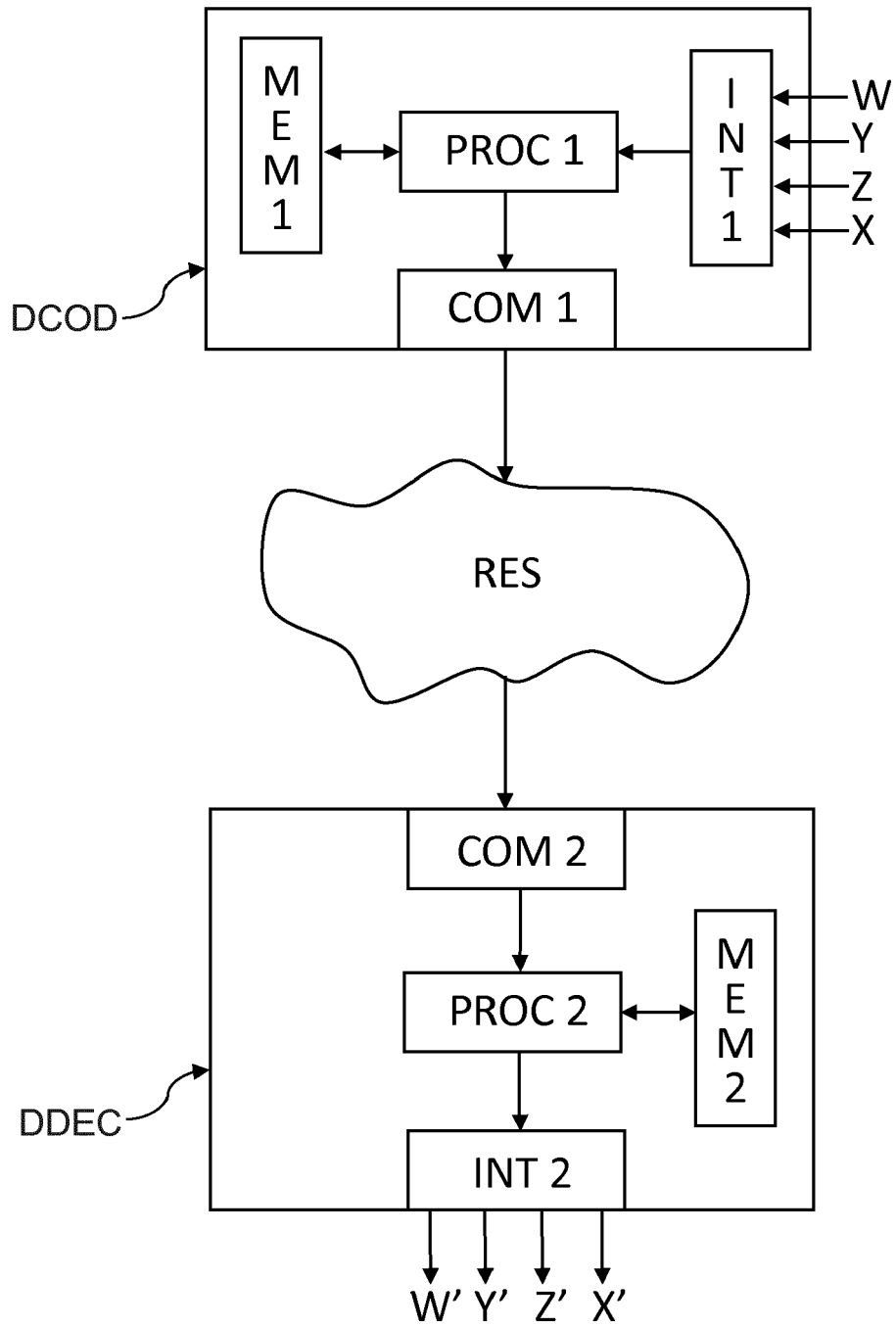


FIG. 7



RAPPORT DE RECHERCHE EUROPEENNE

Numéro de la demande  
EP 19 30 5254

5

10

15

20

25

30

35

40

45

50

55

DOCUMENTS CONSIDERES COMME PERTINENTS			
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendication concernée	CLASSEMENT DE LA DEMANDE (IPC)
X	US 2016/155448 A1 (PURNHAGEN HEIKO [SE] ET AL) 2 juin 2016 (2016-06-02)	12,14,15	INV. G10L19/008
Y	* alinéa [0004]; figure 1 *	1,6,7,	
A	* alinéa [0070] - alinéa [0083] *	10,13 2-5,8,9, 11	
Y	----- ROUMEN KOUNTCHEV ET AL: "New method for adaptive karhunen-loeve color transform", TELECOMMUNICATION IN MODERN SATELLITE, CABLE, AND BROADCASTING SERVICES, 2009. TELSIS '09. 9TH INTERNATIONAL CONFERENCE ON, IEEE, PISCATAWAY, NJ, USA, 7 octobre 2009 (2009-10-07), pages 209-216, XP031573422, ISBN: 978-1-4244-4382-6 * section II, en particulier la colonne de gauche en page 211 * -----	1,6,7, 10,13	DOMAINES TECHNIQUES RECHERCHES (IPC)
			G10L
3 Le présent rapport a été établi pour toutes les revendications			
Lieu de la recherche <b>La Haye</b>		Date d'achèvement de la recherche <b>15 août 2019</b>	Examineur <b>De Meuleneire, M</b>
CATEGORIE DES DOCUMENTS CITES X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire		T : théorie ou principe à la base de l'invention E : document de brevet antérieur, mais publié à la date de dépôt ou après cette date D : cité dans la demande L : cité pour d'autres raisons ..... & : membre de la même famille, document correspondant	

EPO FORM 1503 03.02 (P04C02)

**ANNEXE AU RAPPORT DE RECHERCHE EUROPEENNE  
RELATIF A LA DEMANDE DE BREVET EUROPEEN NO.**

EP 19 30 5254

5 La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche européenne visé ci-dessus.  
Lesdits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du  
Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets.

15-08-2019

Document brevet cité au rapport de recherche	Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 2016155448 A1	02-06-2016	EP 3017446 A1	11-05-2016
		US 2016155448 A1	02-06-2016
		WO 2015000819 A1	08-01-2015
-----			

EPO FORM P0480

Pour tout renseignement concernant cette annexe : voir Journal Officiel de l'Office européen des brevets, No.12/82