

19 RÉPUBLIQUE FRANÇAISE  
INSTITUT NATIONAL  
DE LA PROPRIÉTÉ INDUSTRIELLE  
COURBEVOIE

11 N° de publication :  
(à n'utiliser que pour les  
commandes de reproduction)

3 031 202

21 N° d'enregistrement national : 15 62988

51 Int Cl<sup>8</sup> : G 06 F 3/14 (2016.01), H 04 N 5/445

12 DEMANDE DE BREVET D'INVENTION

A1

22 Date de dépôt : 21.12.15.

30 Priorité : 24.12.14 GB 14231666.

43 Date de mise à la disposition du public de la demande : 01.07.16 Bulletin 16/26.

56 Liste des documents cités dans le rapport de recherche préliminaire : *Ce dernier n'a pas été établi à la date de publication de la demande.*

60 Références à d'autres documents nationaux apparentés :

○ Demande(s) d'extension :

71 Demandeur(s) : GE AVIATION SYSTEMS LIMITED — GB.

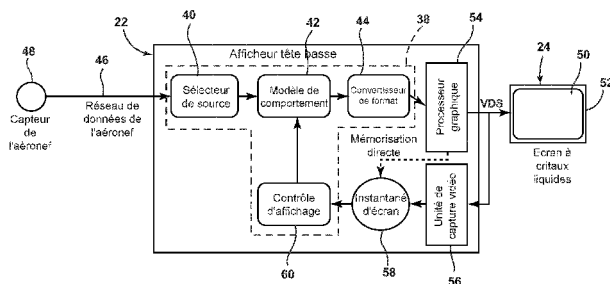
72 Inventeur(s) : CATTON LEWIS WILLIAM.

73 Titulaire(s) : GE AVIATION SYSTEMS LIMITED.

74 Mandataire(s) : CASALONGA & ASSOCIES.

54 SYSTEME ET PROCEDE DE CONTROLE DE L'INTEGRITE DE DONNEES A AFFICHAGE NUMERIQUE.

57 Système et procédé de contrôle d'intégrité de données à affichage numérique dans un système d'affichage (22) ayant un écran comprenant une matrice de pixels avec une zone visible (50), comportant la conversion d'une valeur connue en graphique à code QR, la création d'une image sur l'écran comprenant le dessin du graphique à code QR dans la zone visible (50) de la matrice de pixels, le contrôle du graphique à code QR dessiné sur l'image créée et la détermination de l'intégrité des informations graphiques dessinées.



FR 3 031 202 - A1



## **Système et procédé de contrôle de l'intégrité de données à affichage numérique**

5 Les aéronefs modernes peuvent comporter des systèmes d'affichage pour présenter des informations graphiques à un utilisateur, qui peuvent utiliser un contrôle d'affichage par lequel l'intégrité de l'affichage graphique est déterminée en sélectionnant des données de sortie affichées et en les traitant pour leur rendre  
10 leur forme d'entrée d'origine, à comparer avec ce qui a été entré pour créer les données de sortie. Le contrôle actif de l'affichage est une technique coûteuse.

Dans une première forme de mise en œuvre, l'invention concerne un procédé de contrôle de l'intégrité de données à  
15 affichage numérique dans un système d'affichage ayant un écran comprenant une matrice de pixels avec une zone visible, comportant la conversion d'une valeur connue en graphique à code QR, la création d'une image sur l'écran comprenant le dessin d'un graphique à code QR sur la zone visible de la matrice de pixels, le  
20 contrôle, sur l'image créée, du graphique à code QR dessiné, le décodage du graphique à code QR dessin pour définir des informations décodées, la comparaison des informations décodées avec la valeur connue servant à obtenir le graphique à code QR et la détermination, d'après la comparaison, de l'intégrité des  
25 informations graphiques dessinées.

Dans une autre forme de réalisation, l'invention concerne un système d'affichage ayant un écran d'affichage comprenant une matrice de pixels avec une zone visible, un modèle de comportement délivrant des données à dessiner, un processeur  
30 graphique GPU conçu pour recevoir les données et ayant un signal de sortie qui convertit les données sous une forme graphique sur la

zone visible de l'écran d'affichage pour définir une image affichée ayant un code QR à valeur codée, et une unité de contrôle d'affichage conçue pour traiter les informations de l'image et décoder la valeur du code QR traité sur l'image affichée afin de définir des informations décodées, l'unité de contrôle d'affichage fournissant les informations décodées au modèle de comportement de façon que le modèle de comportement puisse déterminer l'intégrité des informations graphiques dessinées en les comparant avec les données délivrées à dessiner.

10 L'invention sera mieux comprise à l'étude de la description détaillée d'un mode de réalisation pris à titre d'exemple non limitatif et illustré par les dessins annexés sur lesquels :

-la Figure 1 est une vue en perspective d'une partie d'un poste de pilotage comportant un système d'affichage selon une forme de réalisation de l'invention ;

-la Figure 2 est une illustration schématique d'un exemple de système d'affichage selon une forme de réalisation de l'invention ;

-la Figure 3 est un organigramme illustrant un procédé de contrôle d'intégrité de données selon une forme mise en œuvre de l'invention ;

-la Figure 4 est un organigramme détaillant une partie du procédé illustré sur la Figure 3 ;

-la Figure 5 est une illustration schématique d'une partie d'une image qui peut être créée selon le procédé illustré sur la Figure 3 ; et

la Figure 6 illustre schématiquement des techniques de superposition utilisables pour créer une image selon le procédé illustré sur la Figure 3. .

La Figure 1 représente une partie d'un aéronef 10 ayant un poste de pilotage 12. Bien qu'un aéronef commercial ait été représenté, il est envisagé que des formes de mises en œuvre et de

réalisation de l'invention puissent être utilisées dans n'importe quel type d'aéronef classique, par exemple, sans limitation, un aéronef à voilure fixe, à voilure tournante, une fusée, un aéronef personnel, un appareil militaire. Un premier utilisateur (p.ex. un pilote) peut être assis sur un siège 14 sur le côté gauche du poste de pilotage 12 et un autre utilisateur (p. ex. un co-pilote) peut être assis sur un siège 16 sur le côté droit du poste de pilotage 12. Un tableau de bord 18 équipé de divers instruments 20 et d'un système d'affichage 22, qui a été représenté ayant de multiples écrans de vol multifonctions avec des écrans d'affichage 24 qui peuvent être face au pilote et au co-pilote et peuvent fournir à l'équipage de vol des informations d'assistance au vol de l'aéronef 10.

Les écrans de vol à écrans d'affichage 24 peuvent comprendre soit des écrans de vol principaux ou des écrans multifonctions et peuvent afficher toutes sortes d'informations sur l'aéronef, le vol, la navigation et autres, servant au fonctionnement et au pilotage de l'aéronef 10. Les écrans de vol à écrans d'affichage 24 peuvent être à même de présenter du texte et des graphiques en couleur à un utilisateur. Les écrans de vol à écrans d'affichage 24 peuvent être agencés de n'importe quelle manière, notamment en ayant un nombre plus ou moins grand d'écrans et ne doivent pas forcément être dans le même plan ni avoir les mêmes dimensions. Un afficheur tactile ou une surface d'affichage tactile peut faire partie de l'écran de vol à écrans d'affichages 24 et peut être utilisé par un ou plusieurs membres de l'équipage de vol, dont le pilote et le co-pilote, pour interagir avec les systèmes de l'aéronef 10. Par ailleurs, un ou plusieurs dispositifs de commande à curseur 26, tels qu'une souris, et un ou plusieurs claviers multifonctions 28 peuvent être inclus dans le poste de pilotage 12 et peuvent aussi être utilisés par un ou plusieurs membres de

l'équipage de vol, dont le pilote et le co-pilote, pour interagir avec les systèmes de l'aéronef 10.

Un automate 30 peut coopérer avec des organes de l'aéronef 10, dont les écrans de vol à écrans d'affichages 24, les dispositifs de commande à curseur 26 et les claviers 28. L'automate 30 peut  
5 comprendre, entre autres, une mémoire 32 et un processeur 34. La mémoire 32 peut comprendre une mémoire vive (RAM), une mémoire morte (ROM), une mémoire flash, ou encore un ou plusieurs types de mémoire électronique portative différents tels  
10 que des disques, des DVD, des CD-ROM, etc., ou toute combinaison adéquate de ces types de mémoire. Le processeur 34 peut exécuter n'importe quels programmes appropriés pour mettre en œuvre une interface utilisateur graphique (GUI) et un système d'exploitation. Ordinairement, ces programmes comprennent un pilote de  
15 périphériques qui permet à l'utilisateur de s'acquitter de fonctions telles que le choix d'options, la saisie d'instructions et d'autres données, la sélection et l'ouverture de fichiers et le déplacement d'icônes. L'automate 30 peut faire partie d'un SGV ou peut coopérer avec le SGV.

20 Une base de données d'informations, interrogeable par ordinateur, peut être stockée dans la mémoire 32 et être accessible au processeur 34. Le processeur 34 peut exécuter une série d'instructions exécutables pour afficher la base de données ou accéder à la base de données. Selon une autre possibilité, l'automate  
25 30 peut coopérer avec une base de données d'informations. Par exemple, une telle base de données peut être stockée dans un autre ordinateur ou automate. La base de données peut être n'importe quelle base de données adéquate, dont une unique base de données ayant de multiples jeux de données, de multiples bases de données  
30 distinctes en lien les unes avec les autres, voire une simple table de

données. L'automate 30 peut aussi être connecté à d'autres automates (non représentés) de l'aéronef 10.

La Figure 2 représente schématiquement un exemple de système d'affichage 22 utilisable dans l'aéronef 10. Un microprocesseur 38 conçu pour délivrer des données peut être inclus dans le système d'affichage 22. Le microprocesseur 38 peut être n'importe quel processeur adéquat, polyvalent ou spécifique, conçu pour la tâche, ce qui inclut le fait que le microprocesseur puisse être l'automate 30 ou puisse faire partie de l'automate 30. Entre autres, un sélecteur 40 de source, un modèle de comportement 42 et un convertisseur 44 de format peuvent être inclus dans le microprocesseur 38. Les données de l'aéronef peuvent être envoyées au système d'affichage 22 par l'intermédiaire d'un réseau 46 de données d'aéronef ou d'une autre entrée appropriée. Les données de l'aéronef peuvent être issues de n'importe quelle source appropriée, dont un capteur 48 de l'aéronef. Le sélecteur de source 40 peut sélectionner la source des données à afficher, et également déterminer tous les paramètres d'entrée et leur validité d'après n'importe quel nombre de sources. Il est également envisagé que le sélecteur 40 de source puisse délivrer un dictionnaire de données. Un dictionnaire de données est une collection de paires <clef, valeur>. Un dictionnaire de données facilite le partage de données entre de multiples objets à l'aide d'une API au moment d'une compilation. De la sorte, le contenu du dictionnaire de données ne peut être connu que pendant l'exécution.

Le modèle de comportement 42 peut utiliser ces valeurs présentes dans le dictionnaire pour actualiser ses modèles de comportement et générer des données de sortie à dessiner. Le modèle de comportement 42 est une application de la logique métier servant à traiter les données sources et délivrer des données traitées utilisées par le convertisseur 44 de format. Par exemple, cela peut

se faire à l'aide d'une machine d'état ou d'un algorithme de logiciel qui modélise le comportement attendu des divers artefacts graphiques convertis par le convertisseur 44 de format. Le convertisseur 44 de format contient un modèle graphique des artefacts à convertir. Le convertisseur 44 de format actualise le modèle à l'aide de la sortie de données traitées du modèle de comportement 42. Le convertisseur de format 44 convertit son modèle en instructions de dessin. Le processeur graphique (GPU) 54 convertit les instructions de dessin en matrice de pixels affichée dans la zone visible 50. Toutes les unités d'affichage n'utiliseront pas de GPU. En l'absence d'utilisation de GPU, le convertisseur de format 44 peut utiliser un logiciel de GPU hébergé dans le microprocesseur 38.

L'écran d'affichage 24 peut comprendre une matrice de pixels avec une zone visible 50 et une éventuelle zone non visible 52. Un processeur graphique (GPU) 54 peut aussi être inclus et le GPU 54 peut être conçu pour recevoir les données délivrées par le microprocesseur 38 et convertir graphiquement les informations sur la zone visible 50 de l'écran d'affichage 24. De la sorte, le GPU 54 convertit les instructions logicielles fournies par le microprocesseur 38 en pixels physiques sur l'écran. Le GPU 54 peut produire un écran de pixels complet, par l'intermédiaire d'un signal différentiel basse tension (LVDS), à la zone visible 50 de l'écran d'affichage 24. Bien que le GPU 54 ait été représenté séparé du microprocesseur 38, le GPU 54 peut aussi faire partie du microprocesseur 38. Par exemple, le GPU 54 peut être un logiciel dans le microprocesseur 38.

Par ailleurs, une unité de capture vidéo facultative 56 peut être incluse et peut être conçue pour reconverter le LVDS en ensemble de valeurs de pixels dans la mémoire. Cette unité de capture vidéo 56 peut comprendre un FPGA (Réseau prédiffusé

programmable par l'utilisateur). Il est envisagé que le GPU 54 puisse permettre un accès direct à la mémoire du GPU 54 et que, dans de tels cas, une copie directe puisse être utilisée et qu'une unité de capture vidéo soit inutile. Que l'unité de capture vidéo 56 soit utilisée ou non, un instantané de l'image créée sur l'écran d'affichage peut être pris, comme illustré en 58.

Une unité de contrôle 60 d'affichage peut traiter l'instantané de l'image créée pour vérifier que les instructions graphiques envoyées au GPU 54 ont réellement été dessinées correctement sur la zone visible 50 de l'écran d'affichage 24. Bien que l'unité de contrôle d'affichage 60 ait été représentée comme faisant partie du microprocesseur 38, l'unité de contrôle 60 d'affichage peut aussi être séparée. Plus particulièrement, si le texte "123" est sur l'instantané, l'unité de contrôle 60 d'affichage peut reconvertir les pixels en valeur "123". L'unité de contrôle d'affichage 60 peut également vérifier l'emplacement et la rotation des informations. Ces informations peuvent ensuite être fournies au modèle de comportement 42 pour une détermination d'intégrité.

Pendant le fonctionnement, le contrôle d'affichage peut être utilisé par le système d'affichage 22 pour vérifier que les paramètres critiques de l'aéronef sont correctement présentés au pilote. Le système d'affichage 22 peut utiliser la chaîne de rendu suivante, traiter les paramètres d'entrée/sortie, traiter le modèle de comportement et dessiner le graphique sur l'écran d'affichage 24. De la sorte, le microprocesseur 38 et le GPU 54 produisent divers affichages graphiques sur le/les écran(s) d'affichage 24. Lorsque l'application convertit des graphiques, elle peut utiliser un graphique à code QR pour coder une valeur d'un paramètre critique de l'aéronef sur l'image convertie. Ainsi, le microprocesseur 38 utilise des codes QR pour coder des informations placées sur l'image finalement affichée.

L'unité de contrôle d'affichage 60 recherche le/les code(s) QR sur l'image créée par le pipeline de rendu. Par exemple, l'unité de contrôle d'affichage 60 peut comprendre un logiciel qui recherche des données identifiant le code QR dans les données d'affichage, puis traite le code QR. Plus particulièrement, l'unité de contrôle d'affichage 60 peut rechercher un bit de démarrage et un bit d'arrêt pour le code QR et tous les bits de données intermédiaires forment la grille pour le code QR. Selon une autre possibilité, l'unité de contrôle d'affichage recherche à un endroit prédéfini un code QR qui, lorsqu'il est décodé, indique où se trouvent tous les autres codes QR. L'unité de contrôle d'affichage 60 traite l'image et décode le contenu du code QR. Par exemple, l'unité de contrôle d'affichage 60 peut soumettre le code QR à un programme logiciel de décodage.

Par ailleurs, il est envisagé que le graphique à code QR puisse être amené à tourner avec les graphiques que voit le pilote. Il est également envisagé que les codes QR puissent avoir des dimensions variables en fonction des données codées qu'ils contiennent. Quand l'unité de contrôle d'affichage 60 trouvera le graphique à code QR, elle connaîtra l'angle de rotation et les dimensions du graphique à code QR. Plus particulièrement, à mesure que l'unité de contrôle d'affichage lit la mémoire tampon de graphiques ou le flux de signaux LVDS, les données concernant le code QR peuvent être lues pour déterminer l'angle de rotation, les dimensions et l'emplacement du graphique à code QR. Le contenu décodé et l'emplacement, les dimensions et la rotation des codes QR peuvent être envoyés au modèle de comportement 42, lequel compare les informations avec celles utilisées pour dessiner sur l'écran 24. Si d'éventuelles erreurs se sont produites pendant la conversion ou le transfert vers le l'écran, cela affectera le graphique à code QR de la même manière que les graphiques que

voit le pilote. Par conséquent, les erreurs peuvent être détectées. Le microprocesseur 38 peut aussi être conçu pour produire une alerte concernant un échec d'un contrôle.

Des détails d'environnements susceptibles de mettre en œuvre des formes de réalisation de l'invention sont présentés pour assurer une compréhension parfaite de la technologie décrite ici. Cependant, pour un spécialiste de la technique, il sera évident que les exemples de formes de réalisation peuvent être mis en œuvre sans ces détails spécifiques et de plusieurs manières différentes. Les exemples de formes de réalisation sont décrits en référence aux dessins. Ces dessins représentent certains détails de formes de réalisation spécifiques mettant en œuvre un module ou un procédé, ou un produit à programme informatique décrit ici. Cependant, les dessins ne doivent pas être interprétés comme imposant d'éventuelles limitations susceptibles d'être présentes sur les dessins. Le procédé et le produit à programme informatique peuvent être respectivement réalisés et placés sur tout support exploitable par ordinateur. Les formes de réalisation peuvent être mises en œuvre à l'aide d'un processeur existant ou par un processeur spécifique prévu à cette fin ou à une autre, ou par un système câblé.

Comme indiqué plus haut, les formes de réalisation décrites ici peuvent comporter un produit à programme informatique comprenant des supports exploitables par ordinateur destinés à contenir ou sur lesquels sont stockées des structures de données ou des instructions exécutables par ordinateur. Ces supports exploitables par ordinateur peuvent être n'importe quels supports existants, accessibles à un ordinateur polyvalent ou spécifique ou à une autre machine à microprocesseur. Par exemple, ces supports exploitables par ordinateur peuvent comprendre une mémoire vive RAM, une mémoire morte ROM, une EPROM, une EEPROM, un CD-ROM ou autres dispositifs de stockage sur disque optique, de

stockage sur disque magnétique ou autres dispositifs de stockage magnétique, ou n'importe quel autre support utilisable pour contenir ou stocker un code de programme voulu sous la forme de structures de données ou d'instructions exécutables par ordinateur, et accessible à un ordinateur polyvalent ou spécifique ou à une autre machine à microprocesseur. Quand des informations sont transmises ou fournies à une machine via un réseau ou une autre connexion de communication (par voie câblée, radioélectrique ou par une combinaison de liaisons câblées ou radioélectriques), la machine perçoit à juste titre la connexion comme étant un support exploitable par ordinateur. Ainsi, toute connexion de ce type est appelée à juste titre support exploitable par ordinateur. Des combinaisons des supports ci-dessus entrent également dans la définition des supports exploitables par ordinateur. Les instructions exécutables par ordinateur comprennent, par exemple, des instructions et des données, qui amènent un ordinateur polyvalent, un ordinateur spécifique ou des processeurs spécifiques à exécuter une certaine fonction ou un certain groupe de fonctions.

Des formes de réalisation seront décrites dans le contexte général d'étapes d'un procédé qui, dans une forme de réalisation, peuvent être mises en œuvre par un produit à programme comprenant des instructions exécutables par ordinateur, telles que des codes de programmes, par exemple sous la forme de modules de programmes exécutés par des machines en réseau. Globalement, les modules de programmes comprennent des routines, des programmes, des objets, des composants, des structures de données, etc., qui ont pour effet technique d'exécuter des tâches particulières ou de mettre en œuvre des types particuliers de données abstraites. Les instructions exécutables par ordinateur, les structures de données associées et les modules de programmes constituent des exemples de codes de programmes pour exécuter des étapes du procédé

présenté ici. L'ordre particulier de ces instructions exécutables ou des structures de données associées représente des exemples d'actions correspondantes pour mettre en œuvre les fonctions décrites en référence à ces étapes.

5 Les formes de réalisation peuvent aussi être mises en œuvre dans des environnements informatiques distribués où des tâches sont exécutées par des dispositifs de traitement locaux et distants en lien les uns avec les autres (soit par des liaisons câblées, des liaisons radioélectriques, soit par une combinaison de liaisons câblées ou radioélectriques) par l'intermédiaire d'un réseau de communication. Dans un environnement informatique distribué, les modules de programmes peuvent être situés dans des dispositifs de mémorisation locaux et distants.

15 Selon une forme de réalisation de l'invention, la Figure 3 illustre un procédé 100, lequel peut servir à contrôler l'intégrité de données à affichage numérique dans un système d'affichage tel que le système d'affichage 22 ou tout système d'affichage ayant un écran comprenant une matrice de pixels avec une zone visible. Le procédé 100 débute en 102 par la conversion d'une valeur connue en graphique à code QR. Les dimensions du graphique à code QR dépendent de deux éléments. Le premier est la taille de chacun des modules ou carrés composant le code QR. S'il se trouve que la taille des modules est en-deçà de la limite de résolution d'une caméra ou d'un autre système de détection, le code QR ne pourra alors pas être lu par le dispositif. Le second consiste en ce que plus on place de données dans un code QR plus on introduit de rangées et de colonnes de modules dans le code QR pour compenser l'accroissement de la charge de données. Comme la mémoire tampon de graphiques ou le flux de signaux LVDS est ce que lit l'unité de contrôle d'affichage 60, la taille du module peut théoriquement être bien plus petite que celle d'un module lu par une caméra. En outre,

la quantité d'informations stockée est très faible ; par conséquent, le code QR réalisé peut être incroyablement petit, un micro-code QR étant notamment utilisable. A titre d'exemple nullement limitatif, la quantité d'informations peut comprendre des informations telles que celles figurant sur le Tableau 1 ci-dessous.

	Nombre d'unités	Virgules décimales	Exemples
Altitude	5	0	13000 m ou 50000"
Vitesse	3	0	999 nœuds
Tangage	2	2	90,0 degrés

Tableau 1. Exemples de valeurs codées

Il est envisagé que la valeur codée puisse être un paramètre critique de l'aéronef ou au moins une partie d'un paramètre critique de l'aéronef. A titre d'exemples nullement limitatifs, ces paramètres critiques de l'aéronef peuvent comprendre une altitude, une vitesse, une assiette et/ou un cap. Ces paramètres sont critiques pour la sûreté de l'aéronef et il faut donc surveiller ces paramètres pour s'assurer de ce qu'aucune information induisant dangereusement en erreur n'est présentée au pilote.

En 104, une image peut être créée sur l'écran d'affichage 24 et celle-ci peut comprendre un dessin du graphique à code QR dans la zone visible de la matrice de pixels. Bien que le code QR puisse être dessiné sur la zone visible 50 de l'écran d'affichage 24, le code QR peut être dessiné de manière à rester invisible pour l'équipage. Par exemple, comme décrit plus en détail ci-après, le code QR peut être dans le canal vert de l'écran, lequel permet difficilement à l'œil humain de distinguer les petites variations de couleur. En 106, l'image créée peut faire l'objet d'un contrôle du graphique à code QR dessiné. Par exemple, dans le système d'affichage 22 décrit plus haut, l'image créée peut être contrôlée par l'unité de contrôle

d'affichage 60 et/ou le microprocesseur 38. En 108, les informations du graphique à code QR dessiné peuvent être décodées, par l'unité de contrôle d'affichage 60, pour définir des informations décodées. L'unité de contrôle d'affichage 60 peut alors communiquer les informations décodées au microprocesseur 38 comprenant le modèle de comportement 42. La comparaison par le modèle de comportement 42 peut également comprendre une comparaison de l'emplacement affiché du graphique à code QR avec un emplacement de référence.

En 110, les informations décodées peuvent être comparées avec la valeur connue servant à convertir le graphique à code QR. En 112, l'intégrité des informations graphiques dessinées peut être déterminée d'après la comparaison. Dans le système d'affichage 22, la comparaison faite en 110 et la détermination faite en 112 sont effectuées par le modèle de comportement 42 du microprocesseur 38, bien que n'importe quel ordinateur adéquat puisse faire la comparaison.

Le procédé 100 de contrôle d'intégrité de données à affichage numérique dans un système d'affichage est souple et le procédé 100 tel qu'il est illustré ne l'est qu'à titre d'exemple. Par exemple, l'ordre des étapes n'est indiqué qu'à titre d'illustration et n'est aucunement destiné à limiter le procédé 100, étant entendu que les étapes peuvent se dérouler dans un ordre logique différent ou que des étapes supplémentaires ou intermédiaires peuvent être ajoutées sans s'écarter des formes de réalisation de l'invention. A titre d'exemple nullement limitatif, bien que le dessin, la comparaison et la détermination aient été décrits comme étant réalisés par les composants du microprocesseur, le procédé peut être exécuté par toute combinaison adéquate de matériel et de logiciel.

Selon un autre exemple, le dessin du graphique à code QR peut comprendre le dessin du graphique à code QR sur ladite au

moins une partie de paramètre critique de l'aéronef. Cela peut impliquer une technique servant à cacher des informations dans le bit de poids faible d'une couleur. Cela peut inclure le dessin du graphique à code QR dans un canal vert de l'écran. On peut utiliser le canal vert car l'œil humain a beaucoup plus de difficultés à distinguer les petites variations de couleur. N'importe quelle technique adéquate peut être utilisée pour coder l'image finale créée avec un code QR invisible. Par exemple, les valeurs de rouge-vert-bleu peuvent être manipulées différemment ; on peut recourir au choix d'une couleur différente, d'un nombre de bits différent, etc., notamment en utilisant le canal rouge ou bleu.

On peut utiliser un unique graphique à code QR ou de multiples graphiques à code QR pour représenter une valeur complète d'un paramètre critique. Un exemple de procédé pour dessiner le texte et le graphique à code QR est illustré sur la Figure 4. Tout d'abord, la valeur critique est reçue en 120 et le code QR approprié est créé en 122. La police représentant la valeur est convertie, en 124, en texture (c'est-à-dire en image) qui contient des caractères visibles. Pour dessiner le texte sur l'écran, le texte est décomposé en caractères individuels. Chaque caractère est dessiné à l'aide d'un rectangle et chaque rectangle affiche la portion de texture qui représente ce caractère. Dans ce procédé, on peut n'utiliser qu'un seul code QR pour représenter la valeur complète du paramètre critique. Le code QR peut être créé chaque fois que la valeur change. Un masque peut être mis en place, en 126, afin de ne permettre que le changement du bit de poids faible vert, ce qui rend le code QR invisible pour l'équipage. En 128, le texte est dessiné à l'aide des textures et le code QR est dessiné par-dessus.

Sur la Figure 5, les valeurs mises en évidence par les rectangles 150 montrent les endroits où sont utilisées des textures. Du point de vue de l'utilisateur, seuls les chiffres seront visibles et

le graphique à code QR sera masqué car il sera dans le canal vert de l'écran, qui rend difficile la perception des petits changements de couleur par l'œil humain. L'algorithme de contrôle d'affichage scrute l'image et extrait seulement le graphique à code QR. La Figure 6 illustre la superposition du graphique 152 à code QR et du texte dessiné 154. L'avantage apporté par le fait d'avoir un seul graphique à code QR pour une valeur est qu'un grand graphique à code QR peut être utilisé pour couvrir toute la surface du texte texturé. Les inconvénients comprennent le fait que le graphique à code QR doit être créé lors de l'exécution et que le graphique à code QR nécessite un temps de conversion plus long.

Selon une autre possibilité, un graphique à code QR peut être utilisé pour chaque caractère. Dans ce cas, le graphique à code QR peut être inséré dans la mappe de texture de police. Plus particulièrement, chaque caractère peut contenir un graphique à code QR avec le code hexadécimal du caractère. Cela offre l'avantage que le graphique à code QR peut être créé une seule fois pendant la création, hors ligne, de la mappe de texture de police finale et que, pendant la conversion, il n'y a pas de hausse du niveau de performances pour dessiner les graphiques à code QR. Un inconvénient consiste en ce que les graphiques à code QR doivent être suffisamment petits pour ne pas se chevaucher.

Les effets techniques des formes de réalisation décrites plus haut comprennent le fait que l'intégrité des données à affichage numérique peut être contrôlée, ce qui apporte divers avantages dont le fait que les formes de réalisation ci-dessus facilitent le contrôle d'intégrité de types de systèmes d'affichage numérique utilisant des affichages de format des pixels. Beaucoup de solutions existantes pour le contrôle de l'affichage sont destinées à des applications spécifiques, ce qui signifie que toute modification du rendu des graphiques nécessite une modification similaire du contrôle de

l'affichage. Les formes de réalisation décrites plus haut permettent que le contrôle d'affichage soit indépendant du rendu des graphiques. De la sorte, une fois que la fonction de contrôle d'affichage a été certifiée, elle n'a pas à être modifiée et elle est réutilisable. Les solutions existantes pour le contrôle d'affichage nécessitent que l'écran d'affichage soit connu lorsque l'application dessine les paramètres critiques. Cela permet à l'écran d'affichage d'utiliser des techniques de traitement d'image pour reconvertir les graphiques de l'écran en la valeur qui les a générés. Lorsque surviennent, dans l'application, des changements qui affectent le dessin des paramètres critiques, l'écran d'affichage doit être actualisé pour s'adapter. Cette modification est donc très coûteuse, car il faut certifier à nouveau tout le système. D'autres techniques nécessitent une zone extérieure à l'écran pour dessiner un motif déterminé ce qui fait appel à toutes les fonctions graphiques utilisées par l'application. Cela ne contrôle pas l'image que voit le pilote mais contrôle le motif extérieur à l'écran. On suppose que si une erreur survient dans le rendu des paramètres critiques, la même erreur apparaîtra dans le motif extérieur à l'écran. Cette technique nécessite une conversion supplémentaire, ce qui coûte du temps machine et davantage de mémoire, car l'image finale doit être plus grande que la zone visible.

Les formes de réalisation décrites plus haut résolvent tous ces problèmes. L'utilisation d'un code QR signifie que n'importe quelle information peut être codée. L'écran d'affichage décodera le code QR et renverra le contenu à l'application. Cela permet à l'écran d'affichage d'être indépendant, si bien qu'il n'a pas besoin de connaître les informations qu'il traite. Cela signifie également que l'application de l'écran d'affichage peut être certifiée une seule fois et que toute modification de l'application laissera inchangée l'application de contrôle d'affichage. De plus, l'emplacement et la

rotation du code QR peuvent être conformes aux besoins de l'application. L'application de contrôle d'affichage signale tous les codes QR trouvés sur l'image, avec leur emplacement, leurs dimensions et leur rotation. A la différence des techniques existantes, les formes de réalisation décrites plus haut permettent à une application d'envoyer des données telles que des paramètres critiques réels d'un aéronef, via le pipeline de rendu, puis de les renvoyer à une application pour comparaison.

Dans la mesure où cela n'est pas déjà décrit, les différents détails et structures des diverses formes de réalisation peuvent à volonté être utilisés en combinaison les uns avec les autres. Le fait qu'un détail puisse ne pas être présenté dans toutes les formes de réalisation ne sous-entend pas qu'il ne peut pas l'être, c'est seulement pour des raisons de concision de la description. Ainsi, les divers détails des différentes formes de réalisation peuvent être mêlés et adaptés à volonté pour former de nouvelles formes de réalisation, que les nouvelles formes de réalisation soient expressément décrites ou non. Toutes les combinaisons ou permutations de détails décrits ici sont couvertes par le présent exposé.

### Liste des repères

10	Aéronef	60	Unité de contrôle d'affichage
12	Poste de pilotage	100	Procédé
14	Siège	102	Convertir la valeur connue en graphique à code QR
16	Siège	104	Créer une image
18	Poste de pilotage	106	Contrôler l'image créée
20	Instruments	108	Décoder le graphique à code QR créé
22	Système d'affichage	110	Comparer les informations décodées
24	Ecran de vol	112	Déterminer l'intégrité
26	Dispositif de commande à curseur	122	Recevoir la valeur du paramètre critique
28	Clavier	124	Créer un code QR
30	Automate	126	Dessiner la valeur à l'aide de textures
32	Mémoire	128	Installer un masque
34	Processeur	130	Dessiner le code QR
38	Microprocesseur	150	Textures
40	Sélecteur de source	152	Graphique à code QR
42	Modèle de comportement	154	Texte dessiné
44	Convertisseur de format		
46	Réseau de données		
48	Capteur		
50	Zone visible		
52	Zone non visible		
54	GPU		
56	Unité de capture vidéo		
58	Instantané d'écran		

## REVENDICATIONS

5 1. Procédé de contrôle d'intégrité de données à affichage numérique dans un système d'affichage (22) ayant un écran (24) comprenant une matrice de pixels avec une zone visible (50), le procédé comportant :

la conversion (102) d'une valeur connue en graphique à code QR ;

10 la création (104) d'une image sur l'écran (24), comprenant le dessin du graphique à code QR (152) dans la zone visible (50) de la matrice de pixels ;

le contrôle (106) du graphique à code QR dessiné sur l'image créée ;

15 le décodage (108) du graphique à code QR dessiné afin de définir les informations décodées ;

la comparaison (110), par un ordinateur, des informations décodées avec la valeur connue servant à convertir le graphique à code QR ; et

20 la détermination (112), d'après la comparaison, de l'intégrité des informations graphiques dessinées.

2. Procédé selon la revendication 1, dans lequel la comparaison (110) et la détermination (112) sont réalisées par un modèle de comportement (42) mis en œuvre par un programme informatique.

3. Procédé selon la revendication 2, dans lequel la comparaison (110) comprend en outre une comparaison de l'emplacement affiché du graphique à code QR avec un emplacement de référence.

4. Procédé selon l'une quelconque des revendications précédentes, dans lequel la valeur connue est au moins une partie d'un paramètre critique de l'aéronef (10).

5 5. Procédé selon la revendication 4, dans lequel un seul graphique à code QR (152) sert à représenter une valeur complète d'un paramètre critique.

6. Procédé selon la revendication 4, dans lequel un caractère du paramètre critique de l'aéronef contient un graphique (152) à code QR avec un code hexadécimal du caractère.

10 7. Procédé selon la revendication 4, dans lequel les paramètres critiques de l'aéronef comprennent une altitude, une vitesse, une assiette ou un cap.

8. Procédé selon la revendication 4, dans lequel le dessin du graphique à code QR (152) comprend le dessin du graphique à code QR (152) sur ladite au moins une partie d'un paramètre critique de l'aéronef.

9. Procédé selon la revendication 8, dans lequel le dessin du graphique à code QR (152) comprend le dessin du graphique à code QR (152) dans un canal vert de l'écran (24).

20 10. Système d'affichage (22), comportant :  
un écran d'affichage (24) comprenant une matrice de pixels avec une zone visible (50) ;  
un modèle de comportement (42) délivrant des données à dessiner ;  
25 un GPU (54) conçu pour recevoir les données et ayant un signal de sortie qui convertit graphiquement les données sur la zone visible (50) de l'écran d'affichage (24) pour définir une image affichée ayant un code QR à valeur codée ; et  
une unité de contrôle d'affichage (60) conçue pour  
30 traiter l'image affichée et décoder la valeur du code QR dessiné sur l'image affichée afin de définir des informations décodées ; et

dans lequel l'unité de contrôle d'affichage (60) fournit les informations décodées au modèle de comportement (42) de façon que le modèle de comportement (42) puisse déterminer l'intégrité de l'image affichée en la comparant avec les données délivrées à dessiner.

11. Système d'affichage (22) selon la revendication 10, comportant en outre une unité de capture vidéo (56) conçue pour intercepter le signal de sortie du GPU (54) pour définir l'image affichée.

12. Système d'affichage (22) selon la revendication 10 ou 11, dans lequel l'unité de contrôle (60) accède directement à une mémoire (32) du GPU (54) pour obtenir l'image affichée.

13. Système d'affichage (22) selon l'une quelconque des revendications 10 à 12, dans lequel le modèle de comportement (42) est en outre conçu pour produire une alerte concernant un échec d'un contrôle.

14. Système d'affichage (22) selon l'une quelconque des revendications 10 à 13, dans lequel le modèle de comportement (42) est en outre conçu pour masquer l'écran.

15. Système d'affichage (22) selon l'une quelconque des revendications 10 à 14, dans lequel le modèle de comportement (42) reçoit des informations, sous la forme de données, d'un réseau (46) de données d'un aéronef (10).

16. Système d'affichage (22) selon l'une quelconque des revendications 10 à 15, dans lequel les paramètres critiques de l'aéronef comprennent une altitude, une vitesse, une assiette ou un cap.

17. Procédé sensiblement décrit plus haut en référence aux dessins annexés.

18. Système d'affichage (22) sensiblement décrit plus haut en référence aux dessins annexés.

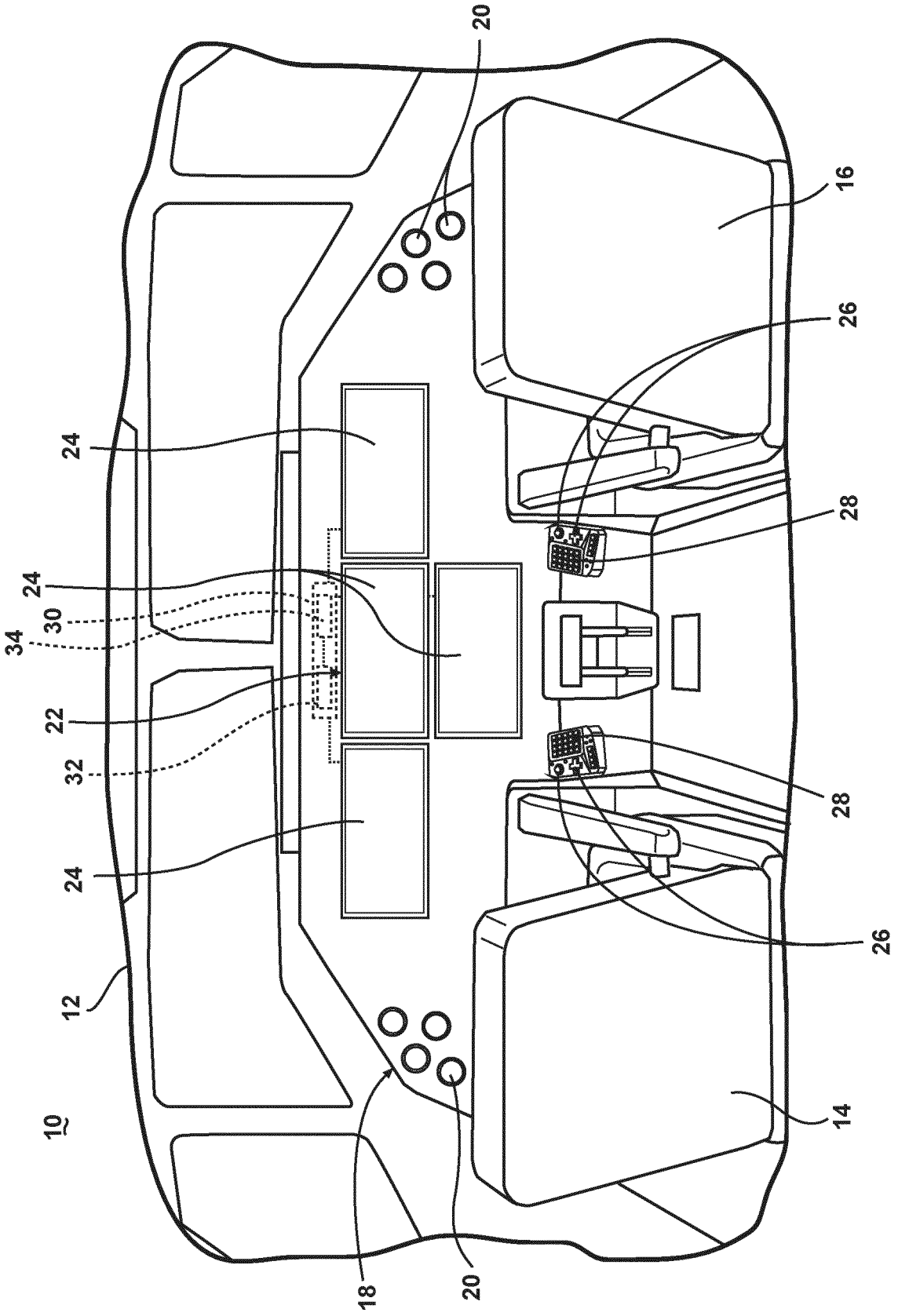


FIG. 1

2/5

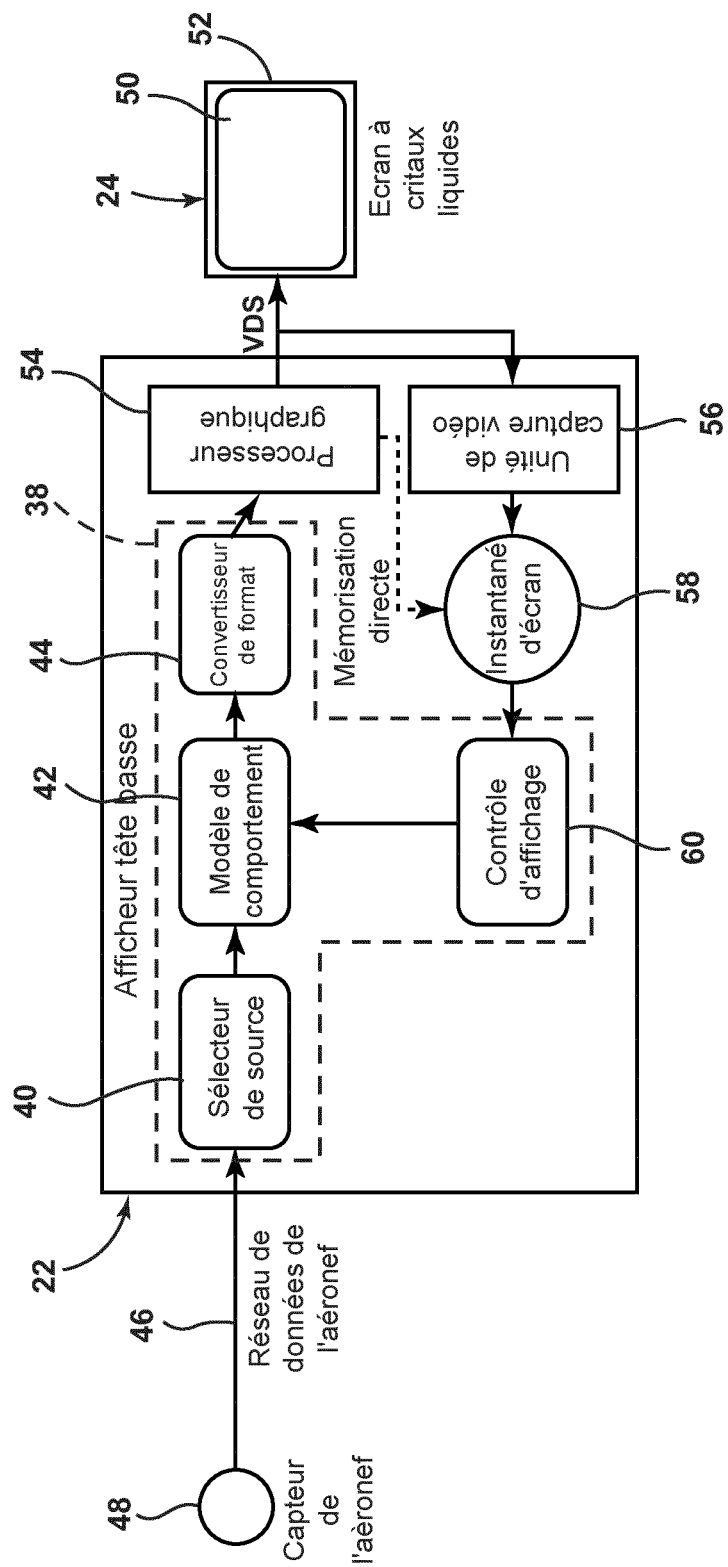


FIG. 2

3/5

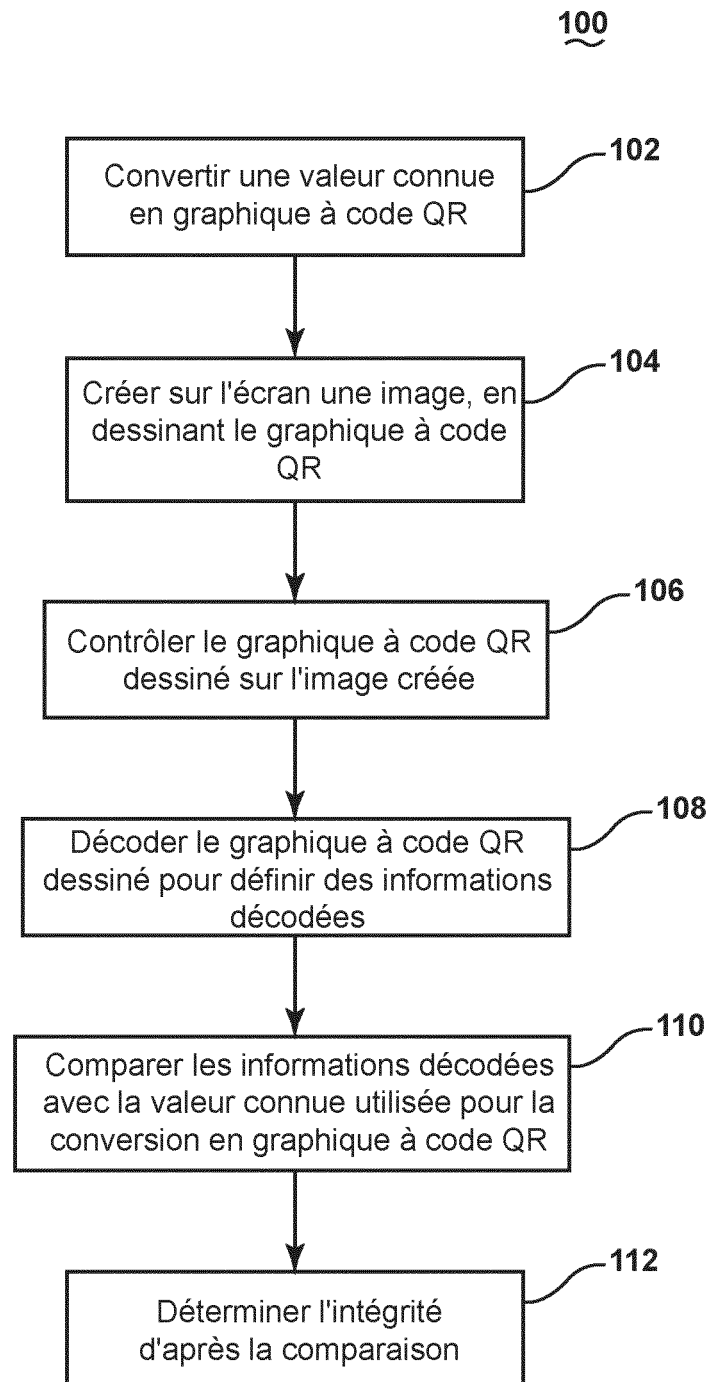


FIG. 3

4/5

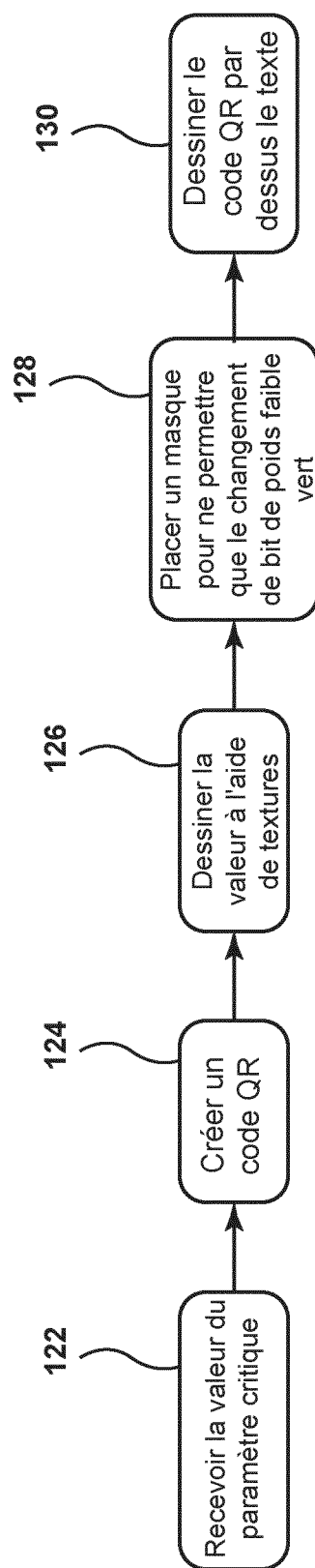


FIG. 4

5/5

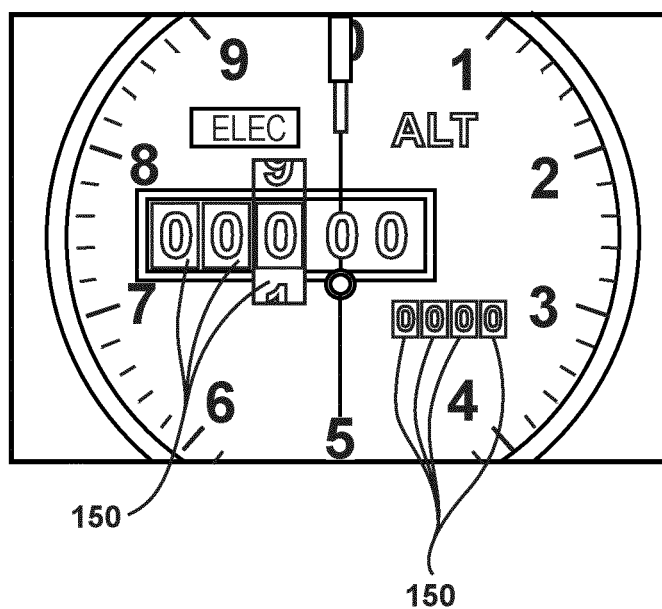


FIG. 5

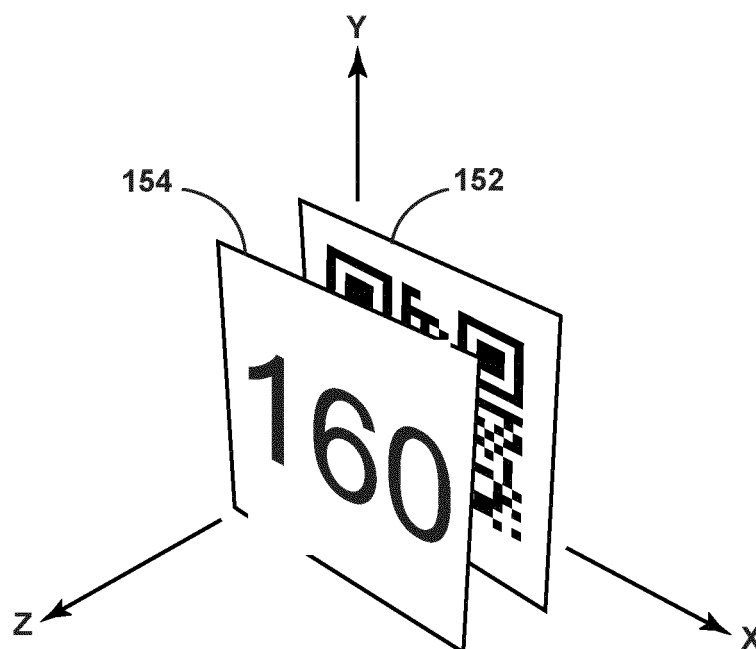


FIG. 6