

(12) DEMANDE INTERNATIONALE PUBLIÉE EN VERTU DU TRAITÉ DE COOPÉRATION
EN MATIÈRE DE BREVETS (PCT)

(19) Organisation Mondiale de la Propriété

Intellectuelle
Bureau international



PCT



(43) Date de la publication internationale
19 octobre 2006 (19.10.2006)

(10) Numéro de publication internationale
WO 2006/108988 A2

(51) Classification internationale des brevets : Non classée

(21) Numéro de la demande internationale :

PCT/FR2006/050323

(22) Date de dépôt international : 10 avril 2006 (10.04.2006)

(25) Langue de dépôt :

français

(26) Langue de publication :

français

(30) Données relatives à la priorité :

0503597

11 avril 2005 (11.04.2005) FR

(71) Déposant (pour tous les États désignés sauf US) :

FRANCE TELECOM [FR/FR]; 6 Place d'Alleray,
F-75015 Paris (FR).

(72) Inventeurs; et

(75) Inventeurs/Déposants (pour US seulement) : BRETON,
Gaspard [FR/FR]; 6 Rue de l'Armor, F-35760 Saint Gre-
goire (FR). CAILLIERE, David [FR/FR]; 32 Bd Laennec,
F-35000 Rennes (FR). AUDOIN, Alexandre [FR/FR]; 34
Rue des Cèdres, F-35150 Janze (FR).

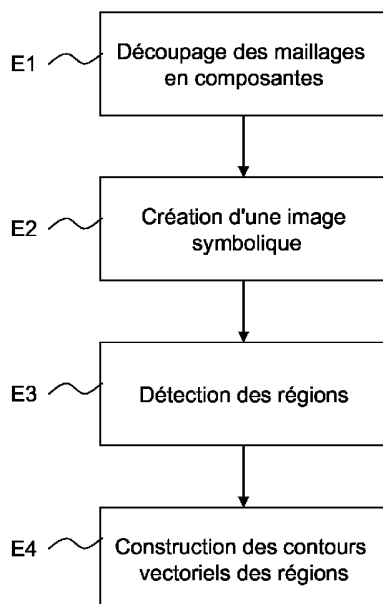
(74) Mandataire : URBILLAC, Chantal; FRANCE TELE-
COM R & D/PIV/BREVETS, 38-40 Rue Du Général
Leclerc, F-92794 Issy Les Moulineaux Cedex 9 (FR).

(81) États désignés (sauf indication contraire, pour tout titre de
protection nationale disponible) : AE, AG, AL, AM, AT,
AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO,
CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB,
GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG,
KM, KN, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, LY,

[Suite sur la page suivante]

(54) Title: METHOD FOR CONVERTING A THREE-DIMENSIONAL IMAGE INTO A TWO-DIMENSIONAL IMAGE

(54) Titre : PROCEDE DE CONVERSION D'UNE IMAGE A TROIS DIMENSIONS EN IMAGE A DEUX DIMENSIONS



E1 CUTTING UP GRIDS INTO COMPONENTS

E2 CREATING A SYMBOLIC IMAGE

E3 DETECTING REGIONS

E4 CONSTRUCTING VECTORIAL CONTOURS OF REGIONS

(57) Abstract: The invention relates to a method for converting a three-dimensional image into a two-dimensional image, characterized by comprising the following steps: determining (E1) independent components of the topological point of view of said three-dimensional image; projecting (E2), onto a surface (Sf) from a point of view (Po), at least a portion of each of the determined components, the depths, in a viewing frustum formed by said point of view, vertexes of the portions of the determined components being preserved in the data of the projected portions of components; first application (E2) of the Z-buffer algorithm to the portions of the components thus projected, and uniformly coloring these projected portions of components, distinctive colors, without correlation with their original colors, being respectively attributed to each of said portions of components in order to obtain a first pixel image.

(57) Abrégé : L'invention concerne un procédé de conversion d'une image à trois dimensions en image à deux dimensions, caractérisé en ce qu'il comporte les étapes de: Détermination (E1) de composantes indépendantes du point de vue topologique, de ladite image à trois dimensions, Projection (E2) sur une surface (Sf)

[Suite sur la page suivante]

WO 2006/108988 A2



MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

- (84) **États désignés** (sauf indication contraire, pour tout titre de protection régionale disponible) : ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), eurasién (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), européen (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Déclaration en vertu de la règle 4.17 :

— relative à la qualité d'inventeur (règle 4.17.iv))

Publiée :

— sans rapport de recherche internationale, sera republiée dès réception de ce rapport

En ce qui concerne les codes à deux lettres et autres abréviations, se référer aux "Notes explicatives relatives aux codes et abréviations" figurant au début de chaque numéro ordinaire de la Gazette du PCT.

depuis un point de vue (Po), d'au moins une partie de chacune des composantes déterminées, les profondeurs, dans une pyramide de vue formée par ledit point de vue, des sommets des parties des composantes déterminées étant conservées dans les données des parties de composantes projetées, Première application (E2) de l'algorithme dit du "Z-buffer" aux parties de composantes ainsi projetées, et coloriage uniforme desdites parties de composantes projetées, des couleurs distinctives, sans corrélation avec leurs couleurs d'origine, étant attribuées à respectivement chacune desdites parties de composantes, afin d'obtenir une première image de pixels.

Procédé de conversion d'une image à trois dimensions en image à deux dimensions

5

La présente invention concerne de manière générale le domaine du traitement d'images, et en particulier les procédés de conversion d'images à
10 trois dimensions en images à deux dimensions.

Les systèmes actuels de traitement d'images à trois dimensions permettent de visualiser des images fixes ou animées d'objets à trois dimensions. Lorsque ces systèmes, ou moteurs graphiques, sont utilisés sur un
15 terminal de faible capacité de calcul de rendu graphique, par exemple sur un terminal mobile, la visualisation, ou rendu, d'une image à trois dimensions sur le terminal utilise d'importantes ressources de calcul, et dégrade la qualité du service qui utilise le moteur graphique du terminal.

Une façon de diminuer les ressources de calcul nécessaires, par
20 exemple lorsqu'un terminal mobile doit afficher des images à trois dimensions reçues depuis un système distant, est de convertir ces images à trois dimensions en images à deux dimensions, plus simples que les rendus correspondants.

Il existe notamment des formats d'images vectorielles à deux dimensions
25 qui permettent une représentation intermédiaire entre une image à trois dimensions et l'image de rendu sur un terminal. Ceux-ci présentent l'avantage de supprimer une partie du calcul sur le terminal, avec un format de représentation réduit par rapport à une image de pixels à deux dimensions. Les systèmes de conversion d'images à trois dimensions en images vectorielles à
30 deux dimensions sont basés sur la projection de chaque polygone ou triangle

de l'image à trois dimensions sur une surface de projection, chaque polygone ou triangle étant projeté un par un. Chaque triangle ou polygone projeté est ensuite codé sous forme vectorielle, les triangles ou polygones adjacents pouvant parfois être fusionnés dans un même contour vectoriel.

5 Ces systèmes présentent néanmoins l'inconvénient d'utiliser beaucoup de place mémoire, puisque chaque triangle ou polygone de l'image initiale est codé sous forme vectorielle. De plus, ces systèmes ne permettent pas de régler efficacement les problèmes de chevauchements entre plusieurs triangles ni de distinguer si un triangle est devant ou derrière un autre triangle lorsque
10 les deux triangles sont intersectés.

La présente invention a pour but de résoudre les inconvénients de la technique antérieure en fournissant un procédé et des dispositifs permettant de
15 convertir une image à trois dimensions en image à deux dimensions, et de coder l'image obtenue en image vectorielle, éventuellement enrichie de traits de pliage. Le procédé de conversion règle les problèmes de chevauchements entre triangles intersectés. De plus l'image vectorielle issue du procédé minimise les ressources mémoires nécessaires à son stockage en fonction de
20 la précision souhaitée. Enfin les images vectorielles issues de l'invention sont utilisables dans une animation en limitant les phénomènes de discontinuité de mouvement entre les images vectorielles successives.

A cette fin, l'invention propose un procédé de conversion d'une image à trois dimensions en image à deux dimensions, caractérisé en ce qu'il comporte
25 les étapes de:

- Détermination de composantes indépendantes du point de vue topologique, de ladite image à trois dimensions,
- Projection sur une surface depuis un point de vue, d'au moins une partie de chacune des composantes déterminées, les profondeurs,
30 dans une pyramide de vue formée par ledit point de vue, des

sommets des parties des composantes déterminées étant conservées dans les données des parties de composantes projetées,

- Première application de l'algorithme dit du "Z-buffer" aux parties de composantes ainsi projetées, et coloriage uniforme desdites parties de composantes projetées, des couleurs distinctives, sans corrélation avec leurs couleurs d'origine, étant attribuées à respectivement chacune desdites parties de composantes, afin d'obtenir une première image de pixels.

L'invention permet avantageusement de s'affranchir des problèmes de chevauchements entre triangles des composantes de l'image à trois dimensions lors de leur projection. L'image en trois dimensions ainsi convertie par le procédé selon l'invention est par exemple utilisée pour être envoyée à un terminal de faible capacité. Ce dernier bénéficie ainsi d'un affichage de l'image à trois dimensions de qualité acceptable tout en utilisant moins de ressources de calcul que s'il calculait lui-même un rendu classique d'image à trois dimensions.

Selon une caractéristique préférée, le procédé comporte l'étape supplémentaire de détection, dans la première image de pixels obtenue à l'issue de l'étape de première application, des contours extérieurs et intérieurs des régions coloriées et indépendantes du point de vue topologique de ladite image de pixels.

Ceci permet de limiter les ressources mémoires nécessaires au codage de l'image de pixels obtenues. Cette économie de ressources mémoires peut s'avérer indispensable pour permettre à un terminal de faible capacité utilisant l'invention de rendre une bonne qualité de service, notamment dans le cas d'un service vidéo.

Selon une caractéristique préférée, le procédé comporte l'étape supplémentaire d'approximation polygonale par subdivision des contours

déTECTÉS à l'issue de l'étape supplémentaire de détection, en des contours vectoriels, afin d'obtenir une image vectorielle à deux dimensions.

5 Ce codage vectoriel simplifie le codage de l'image à deux dimensions obtenue, et permet de diminuer encore les ressources mémoires nécessaires à ce codage.

Selon une caractéristique préférée, chaque contour vectoriel obtenu est orienté suivant un sens de parcours, qui indique si ledit contour vectoriel est un contour intérieur ou un contour extérieur d'une région coloriée de l'image vectorielle à deux dimensions obtenue.

10 Cette orientation obéit à une convention reconnue de codage vectoriel et permet d'obtenir une image vectorielle facilement réutilisable par des formats de type SVG, d'après l'anglais Scalable Vector Graphics.

15 Selon une caractéristique préférée, les contours déTECTÉS à l'issue de l'étape supplémentaire de détection passent entre les pixels limitrophes de chaque région coloriée et indépendante du point de vue topologique de la première image de pixels obtenue à l'issue de l'étape de première application.

20 Ce positionnement des contours permet de dilater les contours extérieurs et de contracter les contours intérieurs, ce qui a l'avantage de supprimer d'éventuels pixels parasites entre deux régions. En effet l'approximation numérique due à l'étape de projection peut faire apparaître des pixels de fond d'image entre régions qui devraient au contraire se toucher.

Selon une caractéristique préférée, le procédé comporte les étapes supplémentaires de:

- 25 - Détection de traits de pliure des composantes de ladite image à trois dimensions,
- Projection desdits traits de pliure sur une surface depuis un point de vue, les profondeurs, dans une pyramide de vue formée par ledit point de vue, des sommets desdits traits de pliure étant conservées dans les données des traits de pliure projetés,

- Deuxième application auxdits traits de pliure de l'algorithme du "Z-buffer", initialisé avec des données issues de l'étape de première application, et coloriage desdits traits de pliure avec une couleur distincte de celles utilisées à l'étape de première application, afin d'obtenir une seconde image de pixels.

5

Le fait de projeter aussi certains traits de pliure importants permet d'obtenir une image à deux dimensions plus réaliste, puisqu'elle met en avant la perspective de certaines composantes de l'image à trois dimensions initiale.

Selon une caractéristique préférée, le procédé comporte les étapes supplémentaires de détection, dans la seconde image de pixels obtenue à l'issue de l'étape de deuxième application, des ensembles distincts de pixels marquant des traits de pliure, et d'approximation polygonale par subdivision desdits traits de pliure en des traits de pliure vectoriels.

10

Le codage des traits de pliure sous forme vectorielle permet de les intégrer dans le format de codage de l'image vectorielle obtenue précédemment, contenant les contours vectoriels de l'image à deux dimensions issue du procédé.

15

L'invention concerne aussi un procédé d'utilisation d'images vectorielles successives obtenues à l'issue de l'étape supplémentaire d'approximation polygonale du procédé de conversion, ledit procédé de conversion étant appliqué à des images en trois dimensions d'une animation pour produire lesdites images vectorielles successives, caractérisé en ce qu'une image vectorielle courante desdites images vectorielles successives est corrigée en utilisant une image vectorielle précédente, afin de limiter les discontinuités de mouvement entre lesdites images vectorielles.

20

25

Cela permet de limiter les discontinuités lors de la visualisation d'une animation utilisant le procédé de conversion d'image selon l'invention.

L'invention concerne aussi un programme d'ordinateur comportant des instructions pour mettre en œuvre les procédés de conversion d'une image à trois dimensions en image à deux dimensions, et d'utilisation d'images

30

vectérielles dans une animation, selon l'invention, lorsque ledit programme est exécuté sur un ordinateur.

L'invention concerne encore un dispositif de conversion d'une image à trois dimensions en image à deux dimensions contenant des moyens de mise en œuvre des procédés précédemment présentés.

L'invention concerne aussi un moteur graphique contenant des moyens de mise en œuvre des procédés précédemment présentés.

Le dispositif de conversion d'une image à trois dimensions en image à deux dimensions et le moteur graphique présentent des avantages analogues à ceux des procédés.

D'autres caractéristiques et avantages apparaîtront à la lecture de modes de réalisation préférés décrits en référence aux figures dans lesquelles :

- la figure 1 représente les différentes étapes du procédé selon l'invention,
- la figure 2 représente les étapes de l'algorithme utilisé lors de la première étape du procédé selon l'invention,
- la figure 3 représente une image à trois dimensions à l'intérieur d'une pyramide de vue formée par l'angle de vue d'un observateur,
- la figure 4 représente une image de pixels obtenue à l'issue de la deuxième étape du procédé selon l'invention,
- la figure 5 représente un triangle d'une image à trois dimensions vu de devant et un triangle d'une image à trois dimensions vu de derrière, depuis le point de vue d'un observateur,
- la figure 6 représente une image de pixels obtenue à l'issue de la deuxième étape du procédé selon l'invention, réalisée selon une première variante,

- la figure 7 représente une image de pixels obtenue à l'issue de la deuxième étape du procédé selon l'invention, réalisée selon une seconde variante,
 - la figure 8 représente les étapes de réalisation de la deuxième étape
 - 5 du procédé selon l'invention, selon une seconde variante,
 - la figure 9 représente un contour extérieur d'une région de pixels,
 - la figure 10 représente les étapes d'un algorithme de détection de contour utilisé lors de la troisième étape du procédé selon l'invention,
 - la figure 11 représente les étapes d'un algorithme d'approximation
 - 10 polygonale par subdivision d'un contour de régions de pixels, utilisé lors de la quatrième étape du procédé selon l'invention,
 - la figure 12 représente un contour d'une région de pixels subdivisé en deux contours,
 - la figure 13 représente un contour d'une région de pixels subdivisé en
 - 15 quatre contours,
 - la figure 14 représente l'utilisation d'une image de pixels agrandie pour dilater des contours extérieurs de régions et contracter des contours intérieurs de régions.
- 20
- Selon un premier mode de réalisation du procédé selon l'invention, les éléments des maillages d'une image à trois dimensions sont convertis en un ensemble de zones à deux dimensions, représentables en minimisant le nombre de vecteurs par contour.
- 25
- Les éléments d'un maillage de l'image à trois dimensions sont décrits en termes de sommets, d'arêtes entre ces sommets, et de triangles ou polygones formés par ces sommets et arêtes. Ces éléments décrivent un ou plusieurs objets à trois dimensions dans l'image qui ont une propriété commune, par exemple une même texture. Un maillage peut donc être représenté sous forme
- 30
- de graphe, dans lequel chaque sommet du graphe est un sommet du maillage,

et dans lequel les arêtes du graphe sont les arêtes qui relient les sommets du maillage. Pour simplifier la description du procédé selon l'invention, on suppose dans la suite que les maillages de l'image à trois dimensions ne comprennent pas de polygones à plus de trois côtés.

5 Il est à noter qu'un maillage d'image à trois dimensions contient aussi de nombreuses autres données sur ses éléments, qui ne seront pas détaillées ici, telles que des données sur leur texture, leur couleur ou leur éclairage.

Dans une première étape E1 du procédé selon l'invention, représentée à la **figure 1**, les maillages de l'image à trois dimensions sont découpés en
10 composantes. Les composantes d'un maillage à trois dimensions sont les éléments du maillage qui ne sont pas reliés de manière topologique entre eux. Par exemple dans un maillage qui associe deux yeux d'un personnage parce qu'ils ont la même texture, chaque œil forme une composante du maillage. Afin d'effectuer ce découpage en composantes, pour chaque maillage de l'image à
15 trois dimensions, le graphe formé par ce maillage est parcouru sommet par sommet, de manière à déterminer les éléments du graphe qui sont indépendants les uns des autres, d'un point de vue topologique.

Les étapes suivantes E2, E3 et E4 du procédé selon l'invention seront détaillées plus loin. Elles correspondent respectivement à:

- 20 - La création d'une image symbolique, correspondant à une visualisation de l'image à trois dimensions suivant le point de vue d'un observateur,
- La détection des différentes régions coloriées de cette image symbolique,
- 25 - Et à une approximation des contours de ces régions par des contours vectoriels.

Toutes ces étapes du procédé selon l'invention peuvent être implémentées dans un moteur de rendu graphique, tel que OpenGL ou DirectX. En particulier, la réalisation de l'image symbolique peut utiliser un moteur de

rendu graphique initialisé de telle sorte que les options d'éclairage et d'anti-crénelage ne soient pas utilisées.

Dans la suite, on suppose que l'invention est implémentée de manière logicielle, à travers différents algorithmes utilisés dans les étapes E1 à E4. Les données des maillages de l'image à trois dimensions initiale sont donc conservées en mémoire et traitées au fur et à mesure par ces algorithmes. Certaines données intermédiaires sont conservées dans des mémoires spécifiques, comme détaillé dans la suite. Néanmoins les autres données intermédiaires, même non explicitement indiquées, pourront être conservées en mémoire jusqu'à l'obtention de l'image à deux dimensions finale issue du procédé selon l'invention.

L'étape E1 est explicitée à travers les étapes de l'organigramme de la **figure 2**, qui sont les suivantes:

La première étape a1 est l'initialisation à zéro d'un compteur appelé "ComposanteNum", qui permet de numéroter toutes les composantes de tous les maillages de l'image à trois dimensions de manière unique. Il est à noter que bien que la numérotation d'une composante de l'image n'inclut pas de référence au maillage auquel appartient cette composante, cette appartenance est conservée dans les données associées à chaque composante obtenue au cours de l'étape E1. Ceci permet à la fin du procédé de recolorier chaque zone de l'image à deux dimensions issue du procédé selon une couleur approchée de celle de la composante à trois dimensions à l'origine de cette zone.

L'étape suivante a2 est la sélection d'un maillage M non déjà parcouru dans l'image à trois dimensions. Par exemple, si la liste des maillages de l'image est copiée dans une variable dans laquelle les maillages parcourus sont supprimés au fur et à mesure, il suffit de sélectionner le premier maillage de cette variable.

L'étape suivante a3 est la vérification du fait que le maillage M sélectionné à l'étape a2 existe bien, c'est-à-dire que tous les maillages de

l'image n'ont pas déjà été parcourus. Si tous les maillages de l'image ont été parcourus, l'étape E1 est terminée, toutes les composantes de l'image étant identifiées par un numéro de composante. Sinon l'étape suivante est l'étape a4.

5 L'étape a4 est la recherche du premier sommet S dans le maillage M sélectionné qui ne soit pas déjà marqué comme appartenant à une composante.

10 L'étape suivante a5 est la vérification de l'existence de ce sommet S recherché à l'étape a4. Si aucun sommet non marqué n'a été trouvé, c'est que le maillage M sélectionné est entièrement parcouru, et l'étape suivante est l'étape a6. Dans le cas contraire, c'est-à-dire si le sommet S existe, l'étape suivante est l'étape a7.

15 L'étape a6 est le marquage du maillage M comme maillage parcouru, par exemple en le supprimant de la liste des maillages, et le passage au maillage suivant non déjà parcouru en retournant à l'étape a2.

 L'étape a7 est le parcours récursif de tous les sommets reliés au sommet S, en les marquant comme appartenant à la composante de numéro "ComposanteNum".

20 L'étape suivante a8, est l'incrémentation d'une unité du compteur "ComposanteNum" pour trouver une éventuelle autre composante non encore identifiée dans le maillage M sélectionné. Pour cela l'étape a8 est suivie de l'étape a4.

25 Une fois les composantes de l'image à trois dimensions identifiées, on crée une image symbolique, intermédiaire entre l'image à trois dimensions et l'image vectorielle à deux dimensions issue du procédé, selon une deuxième étape E2 du procédé selon l'invention, représentée à la **figure 1**. Deux variantes de réalisation de l'étape E2 sont présentées.

Dans une première variante, les composantes de l'image à trois dimensions, représentées à la **figure 3** dans un repère orthonormé (O,x,y,z) , sont projetées sur une surface de projection S_f de repère local (O',X,Y) , depuis le point de vue P_o d'un observateur. On considère un angle de vue correspondant qui forme une pyramide de vue régulière P_v , englobant toutes les composantes de l'image à trois dimensions. Dans l'exemple de la **figure 3**, deux composantes, $Comp1$, qui est une surface rectangulaire, et $Comp2$, qui est une surface triangulaire, forment l'image à trois dimensions. Ces composantes ont une intersection non vide dans l'espace de la pyramide de vue P_v .

Au cours de cette projection, on conserve les coordonnées de profondeur de chacune des composantes dans la pyramide de vue P_v par rapport à la surface S_f . Ainsi la projection du sommet (x_1,y_1,z_1) de la composante $Comp2$ est conservée en mémoire sous la forme (X_1,Y_1,z_1) , où seules les coordonnées (X_1,Y_1) correspondent à cette projection proprement dite, la coordonnée z_1 étant la profondeur initiale de ce sommet dans la pyramide de vue P_v .

Les couleurs de chaque pixel de la surface S_f sont initialisées sans marquage de couleur et enregistrées dans un espace mémoire appelé "frame-buffer". Les profondeurs de chaque pixel de la surface S_f sont initialisées avec une profondeur très supérieure à celle des composantes de l'image à trois dimensions, et sont enregistrées dans un espace mémoire appelé "Z-buffer".

A chaque composante projetée est attribuée une couleur, différente pour chaque composante, et on parcourt ensuite cette composante pixel par pixel sur la surface S_f , suivant l'algorithme classique dit du "Z-buffer": au cours de ce parcours, la profondeur et la couleur d'un pixel de la composante sont mémorisées respectivement dans la mémoire "Z-buffer" et la mémoire "frame-buffer" si la profondeur de ce pixel est moins importante que la profondeur du pixel correspondant enregistré dans la mémoire "Z-buffer". La profondeur du pixel d'une composante est calculée à partir des coordonnées de profondeur

conservées pour chaque composante lors de sa projection sur la surface Sf. Ainsi ne sont conservées dans les mémoires "Z-buffer" et "frame-buffer" que les données correspondant aux pixels les plus proches de l'observateur.

5 On obtient une image symbolique, constituée de pixels, dans laquelle à chaque composante projetée est associée une couleur, et dans laquelle les parties de l'image à trois dimensions non visibles depuis l'observateur ne sont pas représentées. L'image symbolique IS1 obtenue à la fin de l'étape E2 dans l'exemple donné à la figure 3 est représentée à la **figure 4**.

10

 Dans une deuxième variante de réalisation de l'étape E2, on commence par projeter seulement les triangles vus de devant des composantes de l'image à trois dimensions sur la surface Sf, depuis le point de vue Po. On applique ensuite à ces triangles l'algorithme du "Z-buffer", comme dans la première
15 variante précédemment décrite, les triangles de composantes différentes étant coloriés avec des couleurs différentes. Les triangles vus de derrière des composantes de l'image à trois dimensions sont alors projetés à leur tour sur la surface Sf. On applique ensuite à ces nouveaux triangles projetés l'algorithme du "Z-buffer", initialisé avec les données des mémoires "Z-buffer" et "frame-
20 buffer" issues de la première application de l'algorithme du "Z-buffer" sur les triangles vus de devant. Lors de cette deuxième application de l'algorithme du "Z-buffer", les triangles de composantes différentes sont coloriés avec des couleurs différentes, non précédemment utilisées lors de la première application de l'algorithme du "Z-buffer".

25

 Les triangles vus de devant d'une composante se distinguent des triangles vus de derrière de cette même composante par l'orientation des vecteurs normaux associés à chacune des surfaces triangulaires de la composante. Ces vecteurs normaux sont habituellement utilisés par les
30 moteurs graphiques pour calculer l'éclairage sur les différentes surfaces d'un

objet à trois dimensions depuis une source lumineuse. En effet, l'orientation des vecteurs normaux de chacun des triangles d'une composante par rapport à la surface de projection S_f permet de déterminer si un triangle donné est vu de devant ou vu de derrière depuis le point de vue P_o d'un observateur, représenté à la **figure 5**. Ainsi si le vecteur normal à un triangle TrA est dirigé vers l'espace de l'observateur, ce triangle est vu de devant. Si le vecteur normal à un triangle TrB est dirigé de l'autre côté, ce triangle est vu de derrière.

Dans la suite, pour simplifier, les expressions "vu de devant" ou "vu de derrière" pour un triangle d'une composante signifieront respectivement "vu de devant depuis le point de vue P_o " ou "vu de derrière depuis le point de vue P_o ".

Cette deuxième variante de réalisation de l'étape E2 permet d'obtenir une image symbolique dans laquelle les éléments d'une composante vus de devant ont une couleur différente des éléments de cette même composante vus de derrière. Cela permet dans certain cas de mieux visualiser la perspective d'objets à trois dimensions, comme par exemple un ruban dont on voit une partie de l'envers. La **figure 6** représente l'image symbolique IS2 obtenue avec un ruban lorsque cette seconde variante de l'étape E2 est utilisée, et la **figure 7** représente l'image symbolique IS3 obtenue avec ce même ruban lorsqu'on se contente d'utiliser la première variante de l'étape E2.

L'organigramme de la **figure 8** explicite les étapes de la deuxième variante de réalisation de l'étape E2, détaillées ci-après.

La première étape b1 est l'initialisation de variables utiles à la réalisation de cette variante:

- un numéro appelé "CompNum" de la composante à traiter dans l'image à trois dimensions, est initialisé à zéro,

- la couleur de coloriage appelée "CouleurNum", utilisée pendant l'application de l'algorithme du "Z-buffer", est initialisée à un niveau de gris très bas,
- la mémoire "Z-buffer" qui conserve les profondeurs de tous les pixels de la surface de projection Sf, est initialisée à une profondeur maximale pour tous les pixels, par exemple une profondeur infinie,
- la mémoire "frame-buffer", qui conserve les couleurs de tous les pixels de la surface Sf, est également initialisée avec une couleur dite de fond, qui n'est pas utilisée comme couleur de coloriage "CouleurNum", pour tous les pixels.

L'étape suivante b2 est la sélection de la composante c de numéro "CompNum" dans une liste des composantes de l'image à trois dimensions.

- L'étape suivante b3 est la vérification de l'existence de la composante c obtenue à l'étape b2. Si la composante c n'existe pas, c'est que toutes les composantes de l'image à trois dimensions ont été traitées. Dans ce cas l'étape suivante est l'étape b9, sinon l'étape suivante est l'étape b4.

- L'étape b4 est la détermination des triangles vus de devant de la composante c. Cette détermination utilise les vecteurs normaux associés à ces triangles, comme expliqué précédemment.

- L'étape suivante b5 est la projection des triangles obtenus à l'étape b4 sur la surface Sf de projection, par rapport au point de vue Po. Les profondeurs des sommets de ces triangles sont conservées dans les coordonnées des triangles projetés.

- L'étape suivante b6 est l'application de l'algorithme du "Z-buffer" aux triangles projetés à l'étape b5, chaque triangle étant colorié avec le niveau de gris "CouleurNum".

- L'étape suivante b7 est l'incrémentement d'une unité du numéro "CompNum" de la composante à traiter, pour considérer une composante suivante.

L'étape suivante b8 est de même l'incrémentation d'une unité du niveau de gris "CouleurNum". L'étape suivante est de nouveau l'étape b2, pour traiter une autre composante, tous les triangles vus de devant de toutes les composantes de l'image à trois dimensions devant être projetés puis soumis à l'algorithme du Z-buffer avant de passer à l'étape b9.

L'étape b9 est la réinitialisation à zéro du numéro "CompNum" de la composante à traiter. Il est à noter que la couleur "CouleurNum" à appliquer lors de l'algorithme du "Z-buffer" n'est pas réinitialisée, puisque les éléments des composantes vus de derrière ne seront pas coloriés de la même couleur que ceux vus de devant. A cette étape, toutes les composantes de l'image à trois dimensions sont déjà passées par les étapes b4 à b6. La mémoire "Z-buffer" ainsi que la mémoire "frame-buffer" contiennent les données de profondeurs et de couleurs des pixels des éléments vus de devant de l'image à trois dimensions, projetés sur la surface Sf.

L'étape suivante b10 est la sélection de la composante c de numéro "CompNum" dans la liste des composantes de l'image à trois dimensions.

L'étape suivante b11 est la vérification de l'existence de la composante c. Si la composante c n'existe pas, c'est que toutes les composantes de l'image à trois dimensions ont été traitées. Dans ce cas l'étape E2 suivant la deuxième variante de réalisation est terminée, et la mémoire "frame-buffer" contient l'image symbolique faite de pixels coloriés avec les différents niveaux de gris "CouleurNum" utilisés. Sinon l'étape suivante est l'étape b12.

L'étape b12 est la détermination des triangles vus de derrière de la composante c. Cette détermination utilise les vecteurs normaux associés à ces triangles, comme expliqué précédemment.

L'étape suivante b13 est la projection des triangles obtenus à l'étape b12 sur la surface Sf de projection, par rapport au point de vue Po. Les profondeurs des sommets de ces triangles sont conservées dans les coordonnées des triangles projetés.

L'étape suivante b14 est l'application de l'algorithme du "Z-buffer" aux triangles projetés à l'étape b13, chaque triangle étant colorié avec le niveau de gris "CouleurNum".

5 L'étape suivante b15 est l'incrémentation d'une unité du numéro "CompNum" de la composante à traiter, pour considérer une composante suivante.

10 L'étape suivante b16 est de même l'incrémentation d'une unité du niveau de gris "CouleurNum". L'étape suivante est de nouveau l'étape b10, pour traiter une autre composante, tous les triangles vus de derrière de toutes les composantes de l'image à trois dimensions devant être projetés puis soumis à l'algorithme du Z-buffer pour obtenir l'image symbolique.

L'image symbolique obtenue à l'issue de l'étape E2 possède les propriétés suivantes:

- 15
- Elle contient autant ou plus de zones de couleurs différentes que de composantes présentes dans l'image à trois dimensions,
 - Elle contient au plus le même nombre de triangles que de triangles dans l'image à trois dimensions.

20 Cependant la finalité de l'image symbolique obtenue n'est pas d'être visualisée, étant donné, entre autres, qu'elle est constituée de pixels, ce qui est très coûteux en ressources mémoire, et que ses couleurs ne sont pas celles de l'image à trois dimensions d'origine. Un moyen de coder cette image symbolique de manière moins coûteuse en ressources mémoire est d'avoir une description des contours de chaque zone de différente couleur dans l'image
25 symbolique. Ces zones sont dans la suite appelées régions. Cette détection des contours des régions de l'image symbolique est l'objet de l'étape E3, représentée à la **figure 1**.

30 Une région est définie par un ensemble de pixels d'une même couleur de composante, contenu à l'intérieur d'un contour extérieur, et éventuellement à l'extérieur d'un ou plusieurs contours intérieurs si des trous ou d'autres

régions sont présents à l'intérieur de cette première région. Un trou est défini par un contour intérieur d'une région, et peut contenir à la fois des pixels non coloriés, c'est-à-dire n'appartenant à aucune composante, et des pixels coloriés d'autres régions. Suite à l'étape E2, les pixels non coloriés correspondent en effet à la couleur de fond de l'image symbolique. Il est à noter que plusieurs régions de l'image symbolique peuvent avoir la même couleur, par exemple si depuis le point de vue P_o une composante en coupe une autre en deux.

Un contour est défini par une suite de sommets de pixels, et par un sens de parcours de cette suite de pixels. Ainsi un contour extérieur est une suite de sommets de pixels qui parcourt dans le sens inverse du sens trigonométrique le contour extérieur de la région à laquelle il est associé. Un exemple de contour extérieur est donné à la **figure 9**. Le contour extérieur de la région formée des pixels p_{22} , p_{23} , p_{33} et p_{34} , le pixel p_{ij} désignant le pixel situé à la ligne i et la colonne j d'une image symbolique, est la suite de sommets $\{s_{22}, s_{23}, s_{24}, s_{34}, s_{35}, s_{45}, s_{44}, s_{43}, s_{33}, s_{32}\}$, où s_{ij} désigne le sommet supérieur gauche du pixel p_{ij} . Un contour intérieur est une suite de sommets de pixels qui parcourt dans le sens trigonométrique le bord d'un trou dans la région associée à ce contour.

Dans la suite, un contour est décrit comme une liste de sommets supérieurs gauches de pixels $\{s_1, \dots, s_n\}$, qui doit être parcouru suivant l'ordre de cette liste, et où n est un entier supérieur à un. De même une arête orientée (s_{ij}, s_{kl}) de ce contour est l'arête issue du sommet s_{ij} et d'extrémité s_{kl} , où s_{ij} et s_{kl} sont les sommets supérieurs gauches respectivement des pixels p_{ij} et p_{kl} .

Un exemple d'algorithme de suivi du parcours extérieur d'une région utilise comme point de départ une arête orientée (s_{ij}, s_{kl}) appartenant au contour extérieur de cette région. A partir de cette arête, on cherche le sommet s_{mn} tel que l'arête orientée (s_{kl}, s_{mn}) appartienne à un bord de cette région et soit dirigée le plus possible vers la droite par rapport à (s_{ij}, s_{kl}) , sans revenir en arrière. On procède de même à partir de l'arête orientée (s_{kl}, s_{mn}) , et

récurivement on construit une liste de sommets jusqu'à revenir au sommet de départ sij. La liste de sommets ainsi obtenue à la fin de l'algorithme décrit le contour extérieur de la région.

De façon similaire, un exemple d'algorithme de suivi d'un parcours
5 intérieur d'une région utilise comme point de départ une arête orientée (sij, skl) appartenant à ce contour intérieur. A partir de cette arête, on cherche le sommet smn tel que l'arête orientée (skl, smn) appartienne à un bord de la région et soit dirigée le plus possible vers la gauche par rapport à (sij, skl), sans revenir en arrière. On procède de même à partir de l'arête orientée (skl,
10 smn), et récurivement on construit une liste de sommets jusqu'à revenir au sommet de départ sij. La liste de sommets ainsi obtenue à la fin de l'algorithme décrit un contour intérieur de la région.

Pour extraire les contours des régions de l'image symbolique obtenue à l'étape E2, cette image est parcourue ligne par ligne de gauche à droite. Une
15 image tampon est utilisée pour stocker les contours et régions détectés au fur et à mesure du parcours de l'image symbolique. Elle permet notamment de déterminer si un pixel de l'image symbolique a déjà été parcouru, s'il est sur une région déjà détectée, ou si une de ses arêtes appartient à un contour déjà parcouru.

20 Il est à noter que suite à l'étape E2, les couleurs des pixels de l'image symbolique permettent de relier chaque région de l'image tampon à une composante de l'image à trois dimensions.

Les étapes de l'algorithme de parcours de l'image symbolique sont
25 représentées à la **figure 10**:

La première étape c1 est l'initialisation de l'image tampon, afin qu'elle ne contienne ni contours ni régions à cette étape de l'algorithme. Le pixel courant p de parcours de l'image, défini comme situé à la ligne courante "ligne" et à la colonne courante "col", est également initialisé au premier pixel supérieur
30 gauche de l'image. La ligne courante est donc initialisée à la première ligne et

la colonne courante à la première colonne de l'image. Le pixel suivant est défini comme situé sur la même ligne que le pixel courant mais sur la colonne immédiatement à droite de la colonne courante. Il est donc initialisé à la première ligne et à la deuxième colonne de l'image.

5 L'étape suivante c2 est un test sur le pixel courant. Si le pixel courant est d'une couleur différente de celle du pixel suivant, et si l'arête droite du pixel courant n'appartient pas à un contour de l'image tampon, cela signifie qu'un nouveau contour a été rencontré. Dans ce cas l'étape suivante est l'étape c3, sinon l'étape suivante est l'étape c7.

10 L'étape c3 est à nouveau un test sur le pixel courant. Si le pixel courant appartient à une région de l'image tampon, c'est que le contour détecté est un contour intérieur de cette région. Dans ce cas l'étape suivante est l'étape c11. Dans le cas contraire, le contour détecté est un contour extérieur d'une nouvelle région, et l'étape suivante est l'étape c4.

15 L'étape c4 est la création de nouvelles variables R de région et C de contour extérieur, correspondant à la nouvelle région et au nouveau contour détectés aux étapes c2 et c3.

 L'étape suivante c5 est le suivi du contour extérieur C, en utilisant l'algorithme décrit précédemment avec comme arête initiale l'arête supérieure
20 du pixel suivant, orientée vers la droite.

 L'étape suivante c6 est la mise à jour de l'image tampon avec le contour C ainsi parcouru, et avec la région R délimitée par ce contour extérieur C.

 L'étape suivante c7 est un test sur le pixel suivant. Si le pixel suivant est en fin de ligne, on passe à l'étape c8, sinon on passe à l'étape c10.

25 L'étape c8 est à nouveau un test sur le pixel suivant. Si celui-ci est sur la dernière ligne, alors l'image a été entièrement parcourue, l'algorithme de parcours est donc terminé. Sinon l'étape suivante est l'étape c9.

 L'étape c9 est la mise à jour du pixel courant p à la première colonne de la ligne qui suit la ligne courante. La ligne courante, la colonne courante, et le
30 pixel suivant sont également mis à jour en fonction de ce nouveau pixel

courant. L'étape suivante est alors à nouveau l'étape c2, pour continuer le parcours de l'image.

5 L'étape c10 est également la mise à jour du pixel courant p au pixel suivant, mais dans le cas où le pixel courant p n'était pas en fin de ligne. La ligne courante, la colonne courante et le pixel suivant sont mis à jour en fonction du nouveau pixel courant. L'étape suivante est à nouveau l'étape c2, pour continuer le parcours de l'image.

L'étape c11 est la création d'une nouvelle variable de contour intérieur, correspondant au nouveau contour détecté à l'étape c2.

10 L'étape suivante c12 est le suivi du contour intérieur créé à l'étape c11, en utilisant l'algorithme décrit précédemment avec comme arête initiale l'arête gauche du pixel suivant, orientée vers le bas.

15 L'étape suivante c13 est la mise à jour de l'image tampon avec le contour intérieur créé à l'étape c11 et complété à l'étape c12, associé à la région à laquelle appartient le pixel courant.

20 L'étape suivante c14 est un test sur la couleur du pixel courant. Si le pixel courant a la même couleur que le fond de l'image, on poursuit le processus de parcours de l'image en passant à l'étape c7. Sinon cela signifie qu'une nouvelle région vient d'être détectée à l'intérieur du trou précédemment détecté à l'étape c3. L'étape suivante est alors l'étape c4, afin de déterminer la nouvelle région détectée et son contour extérieur associé

25 L'algorithme de parcours de l'image symbolique terminé, l'image tampon contient l'ensemble des régions et contours de l'image symbolique. A la fin de cette étape E3 du procédé, les contours extérieurs obtenus sont dilatés et les contours intérieurs sont contractés afin de faire disparaître d'éventuels pixels parasites ayant la couleur de fond entre les régions de l'image symbolique. En effet les calculs nécessaires à la projection des composantes lors de l'étape E2 induisent des erreurs d'approximation numérique susceptibles de provoquer

30 l'apparition de ces pixels parasites.

Un moyen d'effectuer cette dilatation des contours extérieurs et contraction des contours intérieurs est d'utiliser une image symbolique ISA représentée à la **figure 14**, agrandie d'un pixel de largeur et d'un pixel de hauteur par rapport à l'image symbolique originale ISO. L'image symbolique originale est superposée à cette image agrandie, de manière centrée sur l'image symbolique agrandie. Les contours des régions contenus dans l'image tampon sont ensuite tracés sur l'image symbolique originale. Dans l'exemple de la **figure 14**, l'image tampon contient un contour intérieur CIO et un contour extérieur CEO. Les contours extérieurs sont ensuite tracés sur l'image symbolique agrandie en arrondissant les contours extérieurs de l'image originale aux arêtes externes les plus proches dans l'image agrandie. Les contours intérieurs sont de même tracés sur l'image symbolique agrandie mais en arrondissant les contours intérieurs de l'image originale aux arêtes internes les plus proches dans l'image agrandie. Ainsi le contour extérieur CEO est arrondi au contour extérieur CEA sur l'image symbolique agrandie, tandis que le contour intérieur CIO disparaît sur l'image symbolique agrandie, supprimant ainsi un pixel parasite. Les contours de l'image symbolique agrandie ainsi obtenus sont ensuite réduits au format de l'image symbolique originale pour obtenir de nouveaux contours intérieurs et extérieurs respectivement contractés et dilatés.

Afin de limiter encore les ressources mémoires nécessaires au stockage de l'image, et d'obtenir une image à deux dimensions plus facilement utilisable que l'image tampon, la dernière étape E4 du procédé, représentée à la **figure 1**, permet de transformer l'image tampon en une suite de contours vectoriels de régions. Les couleurs des régions associées à ces contours sont conservées en mémoire afin de produire, par correspondance avec les couleurs initiales des maillages de l'image à trois dimensions, une image à deux dimensions qui

ressemble à la visualisation de l'image à trois dimensions initiale, depuis le point de vue Po.

L'étape E4 utilise l'algorithme représenté à la **figure 11** afin de transformer un contour de l'image tampon, formé de sommets de pixels successifs, en une suite limitée de vecteurs reproduisant le contour suivant la précision souhaitée. Les étapes de cet algorithme sont les suivantes, les quatre premières étapes servant à l'initialisation de l'algorithme:

La première étape d1 consiste à trouver les sommets S_1 et S_2 les plus éloignés d'un contour C_0 de l'image tampon, représenté à la **figure 12**.

L'étape suivante d2 est la construction de deux contours C_1 et C_2 associés respectivement aux vecteurs V_1 et V_2 , tels que V_1 est le vecteur $\overrightarrow{S_1 S_2}$ et V_2 le vecteur $\overrightarrow{S_2 S_1}$. Si le contour C_0 est défini par la suite de sommets $\{s_1, \dots, s_n\}$, tels que:

- n est un entier supérieur à 1
- $S_1 = s_i$ et $S_2 = s_j$, où i et j sont des entiers tels que $1 \leq i \leq j \leq n$

Alors le contour C_1 est défini par la suite de sommets $\{s_1, \dots, s_{i-1}, S_1, S_2, s_{j+1}, \dots, s_n\}$, et C_2 par la suite de sommets $\{S_1, s_{i+1}, \dots, s_{j-1}, \dots, S_2\}$.

L'étape suivante d3 est la création d'un contour vectoriel C_v , initialement vide et qui contiendra à la fin de l'algorithme la suite de vecteurs décrivant C_0 .

L'étape suivante d4 est la création d'une liste de contours à subdiviser, chaque contour de la liste étant associé à un vecteur. Cette liste est initialisée aux contours C_1 et C_2 , associés respectivement aux vecteurs V_1 et V_2 .

L'étape suivante d5 est la sélection dans la liste de contours à subdiviser, d'un contour courant C_i associé à un vecteur courant $\vec{V}_i$. Ce contour courant est par exemple le premier contour de la liste de contours à subdiviser.

L'étape suivante d6 est un test sur l'existence du contour courant C_i . Si ce contour n'existe pas, c'est que la liste de contours à subdiviser est vide, ce qui signifie que l'algorithme est terminé. Sinon l'étape suivante est l'étape d7.

5 L'étape d7 est la détermination dans le contour courant C_i , du sommet S_{Max} appartenant à ce contour courant qui est le plus éloigné du vecteur courant $\vec{V}_i$.

10 L'étape d8 est un test sur la distance D_{max} entre le sommet S_{Max} et le vecteur courant $\vec{V}_i$. Si cette distance est supérieure à un seuil D_{seuil} , qui fixe la précision souhaitée d'approximation du contour C_0 par le contour vectoriel C_v , alors on poursuit la détermination d'autres vecteurs plus proches du contour courant C_i que le vecteur courant $\vec{V}_i$ en passant à l'étape d11. Dans le cas contraire la précision obtenue est acceptable et l'étape suivante est l'étape d9.

15 L'étape d9 est l'ajout au contour C_v de l'inverse $-\vec{V}_i$ du vecteur courant $\vec{V}_i$ associé au contour courant C_i . Le fait d'inverser le vecteur courant $\vec{V}_i$ permet de conserver le sens de parcours initial des sommets du contour C_0 , dans le contour vectoriel final C_v .

20 L'étape suivante d10 est la suppression du contour courant C_i dans la liste de contours à subdiviser. En effet la précision du vecteur courant $\vec{V}_i$ étant suffisante pour approcher le contour courant C_i , celui-ci n'a plus besoin d'être subdivisé. L'étape suivante est l'étape d5, afin de poursuivre l'algorithme avec le traitement des contours restants à subdiviser.

L'étape d11 est la subdivision du contour courant en deux contours C_k et C_l associés respectivement aux vecteurs $\vec{V}_k$ et $\vec{V}_l$ tels que si le contour courant C_i est défini par la suite de sommets $\{s_1, \dots, s_h, \dots, s_m\}$, où:

- 25
- h et m sont des entiers tels que $1 \leq h \leq m$,
 - Le vecteur courant est le vecteur $\overrightarrow{s_m s_1}$,
 - Et S_{Max} est le sommet s_h ,

alors le contour C_k est le contour défini par la suite de sommets $\{s_h, s_{h-1}, \dots, s_2, s_1\}$ et le contour C_l par la suite de sommets $\{s_m, s_{m-1}, \dots, s_{h+1}, s_h\}$, le vecteur $\vec{V}_k$ étant le vecteur $\overrightarrow{s_1 s_h}$ et le vecteur $\vec{V}_l$ étant le vecteur $\overrightarrow{s_h s_m}$.

L'étape suivante d12 est l'ajout des contours C_k et C_l associés
 5 respectivement aux vecteurs $\vec{V}_k$ et $\vec{V}_l$, dans la liste des contours à subdiviser, par exemple en début de liste.

L'étape suivante d13 est la suppression du contour courant C_i dans la liste des contours à subdiviser, puisque ce contour vient d'être remplacé par deux autres contours. L'étape suivante est l'étape d5, les contours à
 10 subdiviser étant traités jusqu'à obtention de vecteurs approchant de manière satisfaisante les contours auxquels ils sont associés.

15 Dans l'exemple de la **figure 13**, à la fin de l'algorithme le contour C_0 a été subdivisé en quatre contours C_3, C_4, C_5 et C_6 , associés respectivement aux vecteurs V_3, V_4, V_5 et V_6 . Le contour vectoriel final C_v correspondant au contour C_0 est donc le contour $\{-V_6, -V_5, -V_4, -V_3\}$.

20 Une fois tous les contours de l'image tampon transformés en contours vectoriels suivant l'étape E4 d'approximation polygonale, on obtient une représentation vectorielle à deux dimensions de l'image à trois dimensions initiale, faite de contours vectoriels de régions, pour chacun desquels le sens de parcours permet de déterminer s'il s'agit d'un contour extérieur ou d'un
 25 contour intérieur de région. Il suffit ensuite de réassocier chaque région de cette représentation à la couleur d'origine du maillage correspondant dans l'image à trois dimensions initiale, pour obtenir une visualisation de l'image à trois dimensions initiale depuis un point de vue d'un observateur sous forme d'une image vectorielle à deux dimensions. Cette image vectorielle est

facilement réutilisable par des logiciels de présentation d'images utilisant le format SVG par exemple, d'après l'anglais Scalable Vector Graphics, et est très peu coûteuse en ressources mémoire.

5 Néanmoins, l'application de l'invention à une animation en images à trois dimensions peut présenter un inconfort visuel dus à des phénomènes de discontinuité, les différentes images vectorielles à deux dimensions obtenues étant calculées indépendamment les unes des autres. La cohérence du mouvement temporel des différents contours vectoriels associés à ces images est en effet affectée par les différentes étapes du procédé de conversion,
10 notamment l'étape d'approximation polygonale, qui ne tient pas compte de la notion de mouvement de l'image traitée dans une animation.

Pour remédier à ce problème de continuité de mouvement des contours vectoriels, une amélioration est apportée à l'invention lorsqu'elle est utilisée dans le cadre d'une animation. En effet dans ce cas, le calcul d'une image
15 vectorielle donnée de l'animation tient compte d'une image vectorielle précédente. Par exemple à l'étape E4, pour adoucir les mouvements des contours vectoriels lors de l'animation, au lieu d'utiliser l'algorithme d'approximation polygonale, on utilise les contours vectoriels de l'image précédente dans l'animation. Chaque contour vectoriel de l'image précédente
20 est appliqué au contour intérieur ou extérieur correspondant dans l'image tampon courante de l'animation, par un procédé d'élasticité: à partir du centre du contour intérieur ou extérieur, on déforme les sommets du contour vectoriel qui lui est superposé en les projetant sur le contour intérieur ou extérieur de l'image courante. Pour chaque région de l'image courante, on obtient ainsi un
25 ou plusieurs contours vectoriels correspondants.

Un contour vectoriel et son contour de région associé forment une suite de contours, assimilable à un résultat intermédiaire de l'algorithme d'approximation polygonale précédemment décrit, c'est-à-dire à une liste de contours à subdiviser. On peut donc utiliser cet algorithme pour corriger le
30 contour vectoriel, l'algorithme étant initialisé avec une liste de contours à

subdiviser contenant cette suite de contours. Ainsi si le contour vectoriel est assez proche du contour qui lui correspond, il est utilisé tel quel dans l'animation, sinon il est subdivisé en utilisant l'algorithme d'approximation polygonale.

5 Cette utilisation des contours vectoriels précédents pour obtenir des contours vectoriels courants permet d'adoucir les mouvements, car les sommets des contours vectoriels se déplacent moins, ce qui améliore la visualisation de l'animation.

10 Selon un deuxième mode de réalisation du procédé selon l'invention, le premier mode de réalisation du procédé selon l'invention décrit précédemment est enrichi par l'ajout, dans l'image à deux dimensions issue du procédé, de traits qui marquent les pliures importantes des composantes de l'image à trois dimensions initiale. Ce deuxième mode de réalisation est intéressant lorsque la
15 détection de ces traits de pliure par le moteur de rendu graphique qui utilise ce procédé n'utilise pas de ressources de calcul excessives.

 Dans ce deuxième mode de réalisation de l'invention, les traits de pliure sont traités de manière indépendante des autres composantes de l'image à trois dimensions initiale, qui est traitée suivant les étapes E1 à E4 décrites
20 précédemment.

 Les traits de pliure importants de l'image à trois dimensions sont tout d'abord identifiés. Un moyen de détecter ces traits de pliure est par exemple de parcourir tous les couples de triangles adjacents à l'intérieur de chacune des composantes identifiées à l'étape E1. Lors de ce parcours, si deux triangles
25 adjacents forment un angle aigu inférieur à un certain seuil au niveau de leur arête commune, alors cette arête est potentiellement un trait de pliure important à prendre en compte. Deux types d'arêtes sont à distinguer:

- Les arêtes de silhouette sont des arêtes qui se partagent deux triangles dont l'un est une face avant et l'autre une face arrière,

c'est-à-dire que les normales de ces triangles pointent dans des sens différents depuis le point de vue Po.

- Les arêtes de pliure sont des arêtes qui se partagent deux triangles formant un angle très prononcé, mais qui sont soit tous deux des faces avant soit tous deux des faces arrières.

5

Seuls les secondes arêtes sont à prendre en compte comme traits de pliure d'une composante ainsi parcourue. En effet les traits de silhouette sont reproduits comme traits de contour après la projection sur la surface Sf lors de l'étape E2 et ne nécessitent donc pas une projection à part.

10

Une fois les traits de pliure détectés, la mémoire "frame-buffer" et la mémoire "Z-buffer" obtenues à la fin de l'étape E2, qui contiennent l'image symbolique, sont recopiées afin d'être utilisées pour le traitement des traits de pliure. Ceux-ci sont projetés sur la surface de projection Sf, depuis le point de vue Po, les profondeurs des sommets qui les composent étant conservées en mémoire, puis traités suivant l'algorithme du "Z-buffer" à partir de la copie des mémoires "Z-buffer" et "frame-buffer" obtenues à la fin de l'étape E2. Les traits de pliure sont coloriés pendant ce traitement avec une couleur distincte de celles utilisées lors de l'étape E2, par exemple en noir. Ceci permet d'obtenir uniquement les parties visibles depuis le point de vue Po, des traits de pliure dans l'image à trois dimensions. A la fin de ce traitement on obtient une image symbolique enrichie de traits de pliure, et distincte de celle utilisée à l'étape E3.

15

20

25

Cette image symbolique enrichie de traits de pliure est ensuite parcourue pixel par pixel afin de coder ces traits sous formes d'ensembles distincts de pixels. Ces suites de pixels sont considérées comme des contours non orientés.

30

Ces contours sont ensuite transformés en traits de pliure vectoriels, de manière indépendante de l'étape E4. A cette fin l'algorithme utilisé à l'étape E4 peut être réutilisé, mais en commençant directement à l'étape d4 représentée à la figure 11. Pour chaque contour à traiter, la liste de contours à subdiviser à

l'étape d4 d'initialisation de l'algorithme contient uniquement ce contour, celui-ci étant associé à un vecteur joignant les pixels qui sont situés aux extrémités du trait de pliure par leurs sommets. Cependant dans le cas où le trait de pliure forme un contour fermé, l'algorithme de l'étape E4 est réutilisé sans
5 modifications.

Les traits de pliure vectoriels sont ensuite ajoutés à l'image vectorielle obtenue à la fin de l'étape E4. A la fin de ce deuxième mode de réalisation du procédé selon l'invention, on obtient, de la même manière que dans le premier mode de réalisation du procédé selon l'invention, une visualisation de l'image à
10 trois dimensions initiale depuis le point de vue d'un observateur sous forme d'une image vectorielle à deux dimensions, mais enrichie de traits de pliure qui permettent de mieux rendre la perspective des objets représentés.

REVENDEICATIONS

- 5 1. Procédé de conversion d'une image à trois dimensions en image à deux dimensions, caractérisé en ce qu'il comporte les étapes de:
- Détermination (E1) de composantes indépendantes du point de vue topologique, de ladite image à trois dimensions,
 - Projection (E2) sur une surface (Sf) depuis un point de vue (Po), d'au
 - 10 moins une partie de chacune des composantes déterminées, les profondeurs, dans une pyramide de vue formée par ledit point de vue, des sommets des parties des composantes déterminées étant conservées dans les données des parties de composantes projetées,
 - Première application (E2) de l'algorithme dit du "Z-buffer" aux parties
 - 15 de composantes ainsi projetées, et coloriage uniforme desdites parties de composantes projetées, des couleurs distinctives, sans corrélation avec leurs couleurs d'origine, étant attribuées à respectivement chacune desdites parties de composantes, afin d'obtenir une première image de pixels.
 - 20
2. Procédé de conversion d'une image à trois dimensions en image à deux dimensions selon la revendication 1, caractérisé en ce qu'il comporte l'étape supplémentaire (E3) de détection, dans la première image de pixels
- 25 obtenue à l'issue de l'étape de première application (E2), des contours extérieurs et intérieurs des régions coloriées et indépendantes du point de vue topologique de ladite image de pixels.
3. Procédé de conversion d'une image à trois dimensions en image à
- 30 deux dimensions selon la revendication 2, caractérisé en ce qu'il comporte

l'étape supplémentaire (E4) d'approximation polygonale par subdivision des contours détectés à l'issue de l'étape supplémentaire de détection (E3) en des contours vectoriels, afin d'obtenir une image vectorielle à deux dimensions.

5 4. Procédé de conversion d'une image à trois dimensions en image à deux dimensions selon la revendication 3, caractérisé en ce que chaque contour vectoriel obtenu est orienté suivant un sens de parcours, qui indique si ledit contour vectoriel est un contour intérieur ou un contour extérieur d'une région coloriée de l'image vectorielle à deux dimensions obtenue.

10

5. Procédé de conversion d'une image à trois dimensions en image à deux dimensions selon l'une quelconque des revendications 2 à 4, caractérisé en ce que les contours détectés à l'issue de l'étape supplémentaire de détection (E3) passent entre les pixels limitrophes de chaque région coloriée et
15 indépendante du point de vue topologique de la première image de pixels obtenue à l'issue de l'étape de première application (E2).

6. Procédé de conversion d'une image à trois dimensions en image à deux dimensions suivant l'une quelconque des revendications 1 à 5, caractérisé en ce qu'il comporte les étapes supplémentaires de:
20

- Détection de traits de pliure des composantes de ladite image à trois dimensions,
- Projection desdits traits de pliure sur une surface (Sf) depuis un point de vue (Po), les profondeurs, dans une pyramide de vue formée par ledit point de vue, des sommets desdits traits de pliure étant conservées dans les données des traits de pliure projetés,
25
- Deuxième application auxdits traits de pliure de l'algorithme du "Z-buffer", initialisé avec des données issues de l'étape de première application (E2), et coloriage desdits traits de pliure
30

avec une couleur distincte de celles utilisées à l'étape de première application (E2), afin d'obtenir une seconde image de pixels.

5 7. Procédé de conversion d'une image à trois dimensions en image à deux dimensions suivant la revendication 6, caractérisé en ce qu'il comporte les étapes supplémentaires de détection, dans la seconde image de pixels obtenue à l'issue de l'étape de deuxième application, des ensembles distincts de pixels marquant des traits de pliure, et d'approximation polygonale par
10 subdivision desdits traits de pliure en des traits de pliure vectoriels.

 8. Procédé d'utilisation d'images vectorielles successives obtenues à l'issue de l'étape supplémentaire d'approximation polygonale du procédé de conversion suivant l'une quelconque des revendications 3 à 7, ledit procédé de
15 conversion étant appliqué à des images en trois dimensions d'une animation pour produire lesdites images vectorielles successives, caractérisé en ce qu'une image vectorielle courante desdites images vectorielles successives est corrigée en utilisant une image vectorielle précédente, afin de limiter les discontinuités de mouvement entre lesdites images vectorielles.

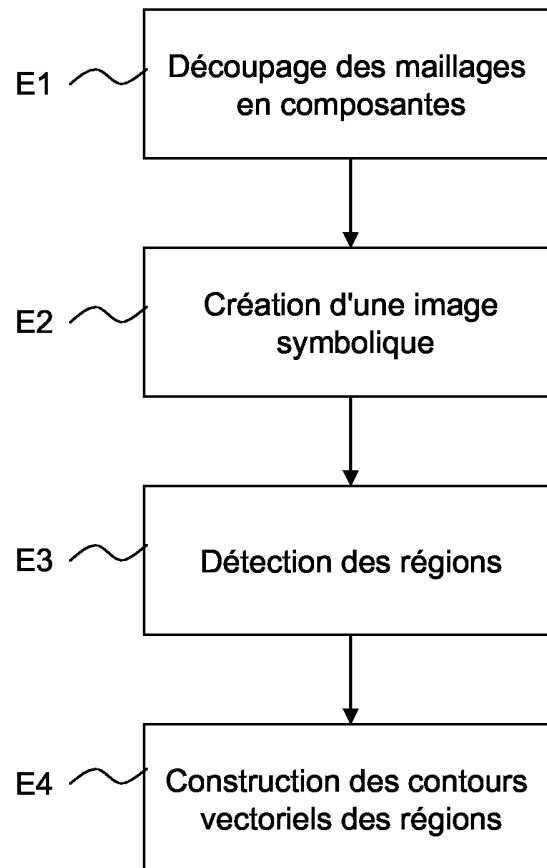
20

 9. Programme d'ordinateur comportant des instructions pour mettre en œuvre le procédé selon l'une quelconque des revendications 1 à 8, lorsqu'il est exécuté sur un ordinateur.

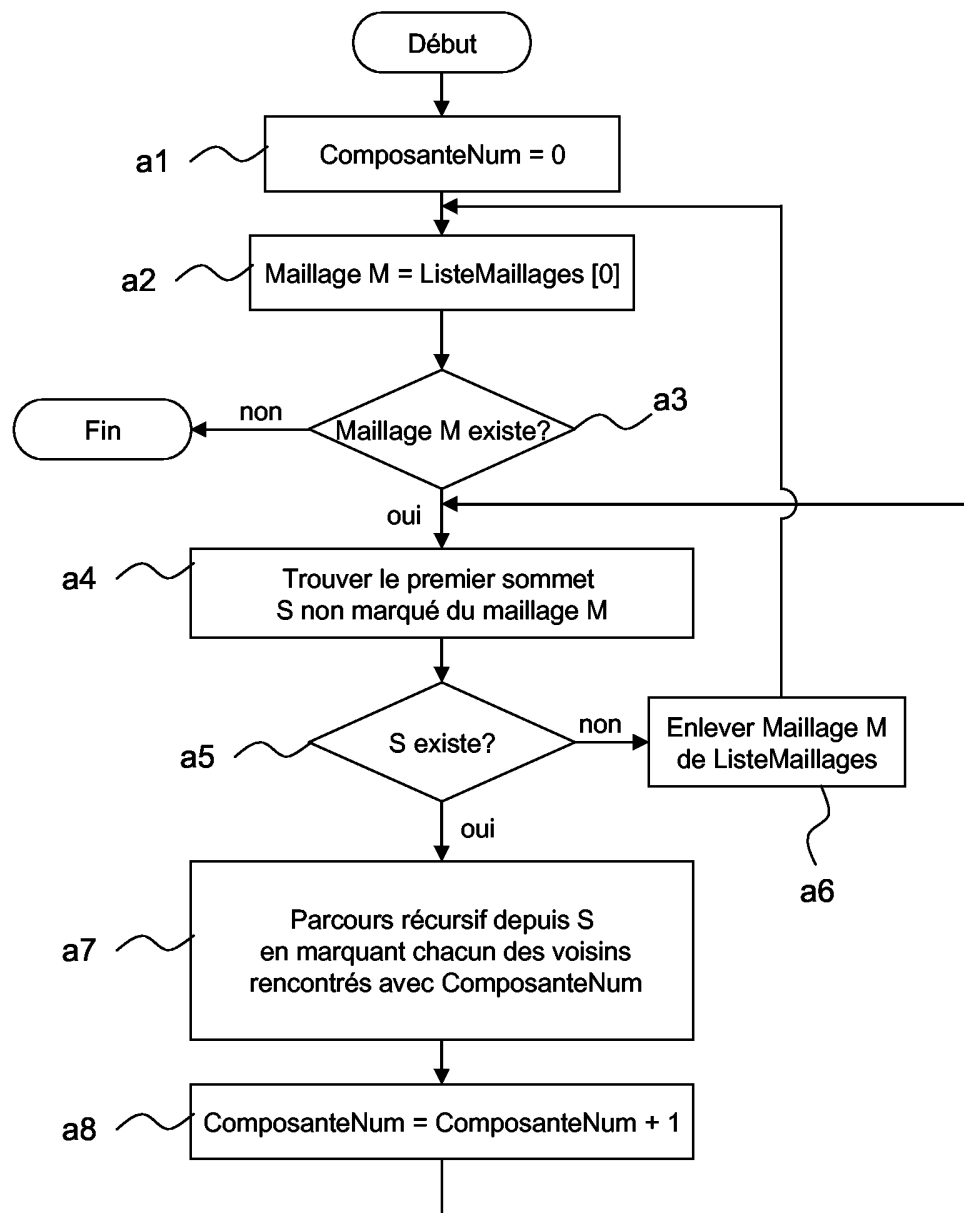
25 10. Dispositif de conversion d'une image à trois dimensions en image à deux dimensions contenant des moyens de mise en œuvre du procédé selon l'une quelconque des revendications 1 à 8.

 11. Moteur graphique contenant des moyens de mise en œuvre du
30 procédé selon l'une quelconque des revendications 1 à 8.

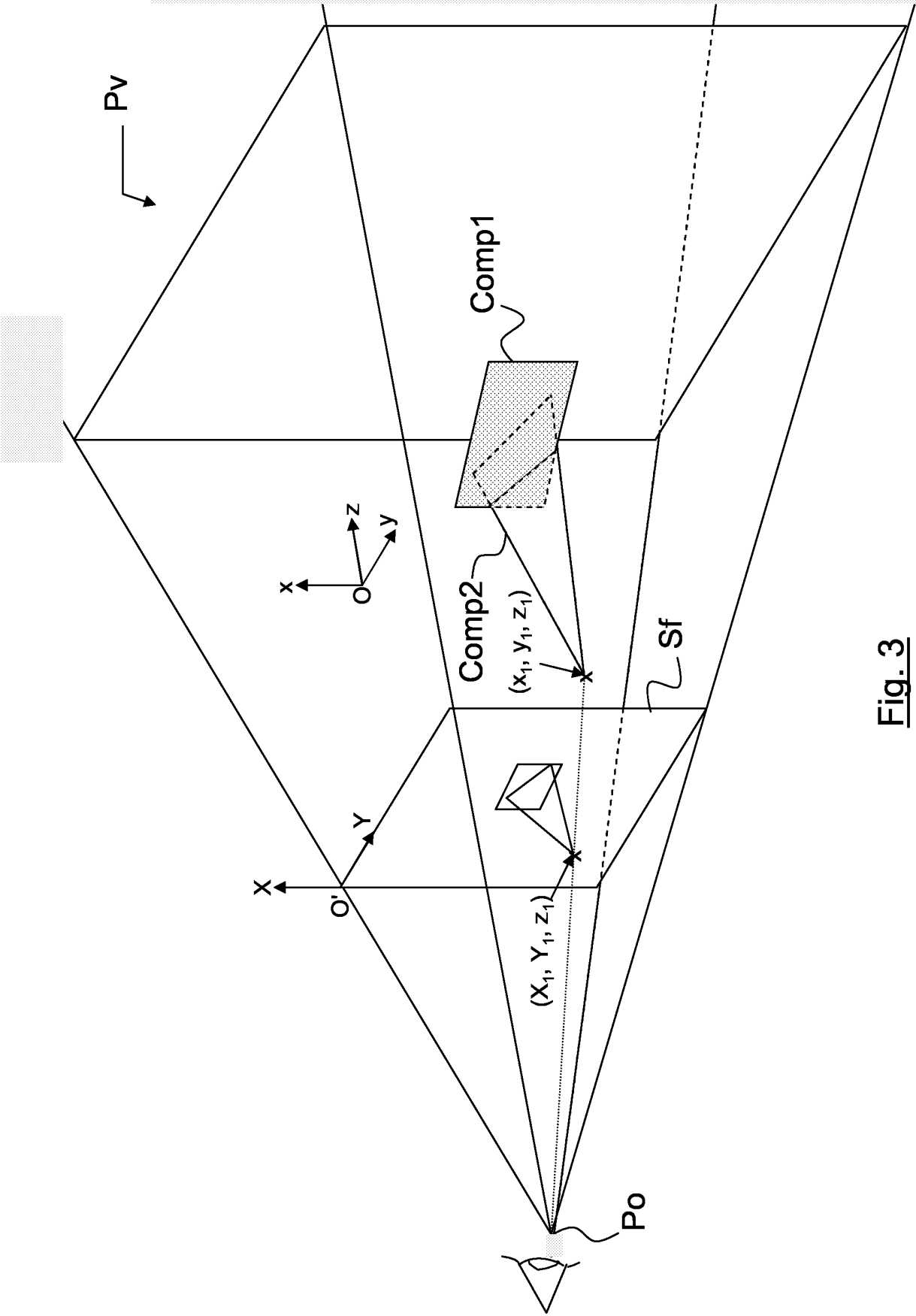
1/10

Fig. 1

2/10

Fig. 2

3/10



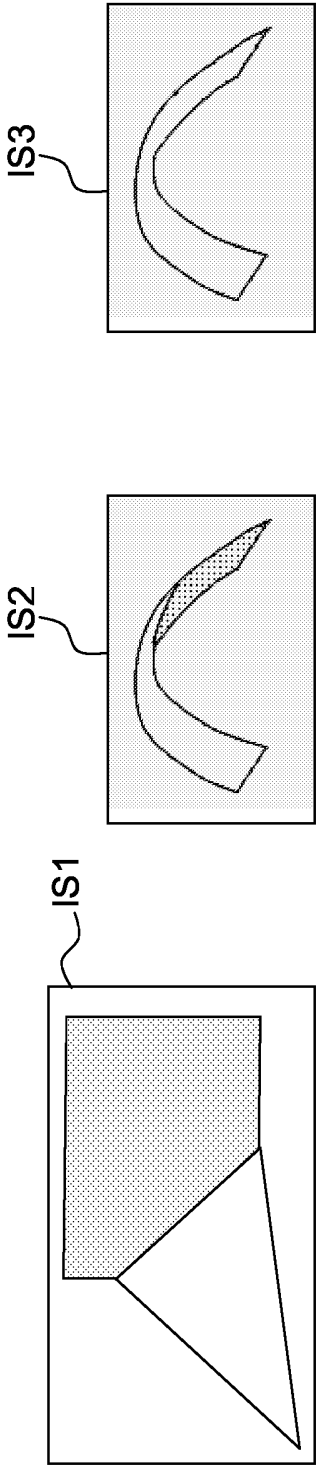


Fig. 7

Fig. 6

Fig. 4

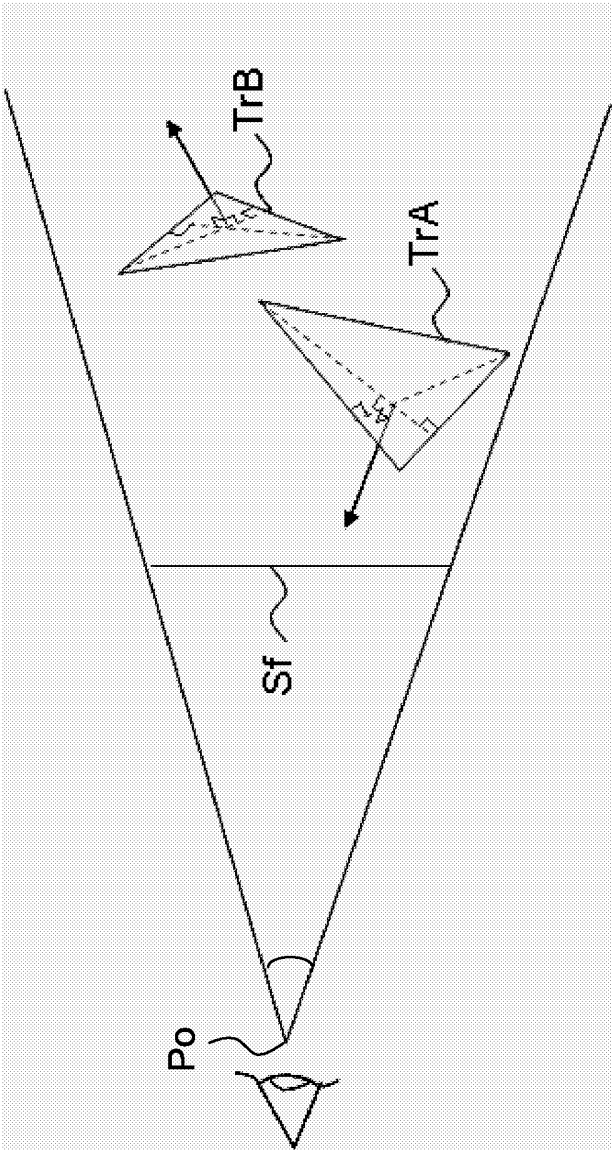


Fig. 5

5/10

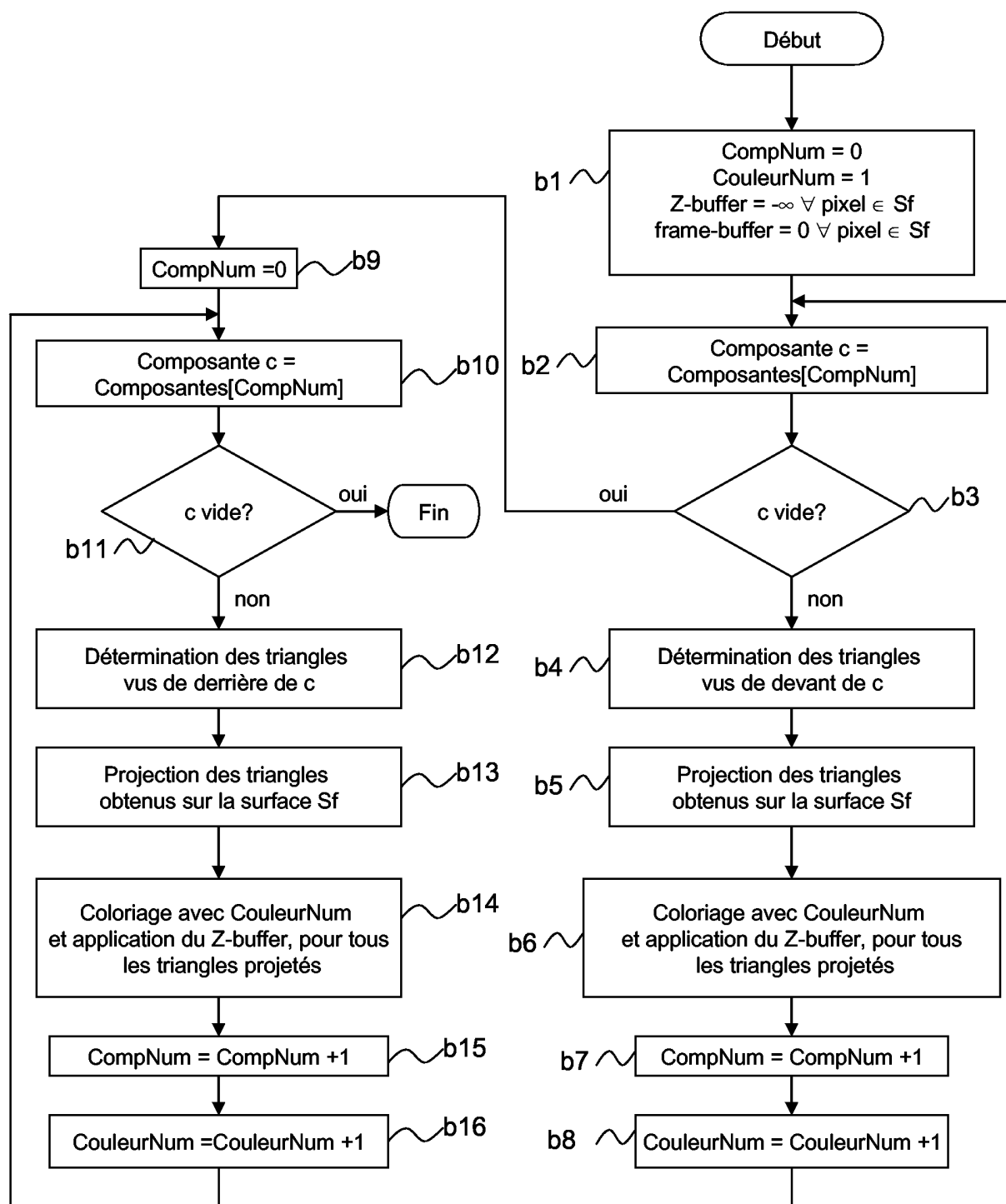


Fig. 8

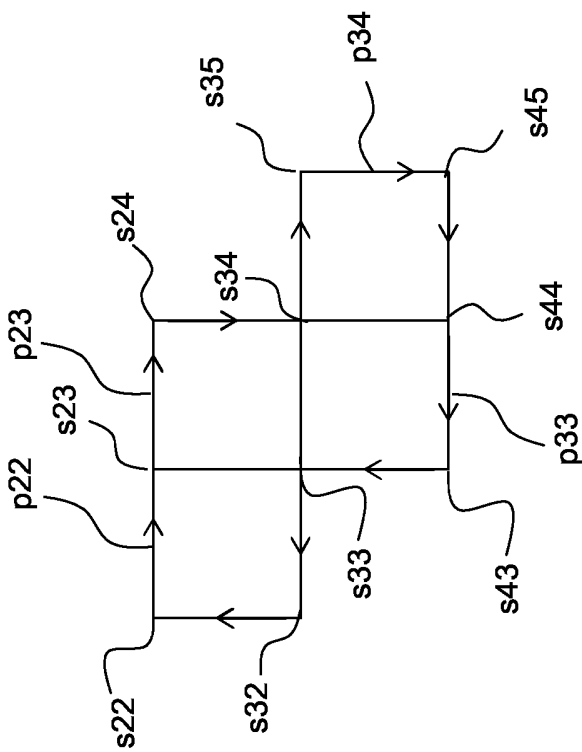


Fig. 9

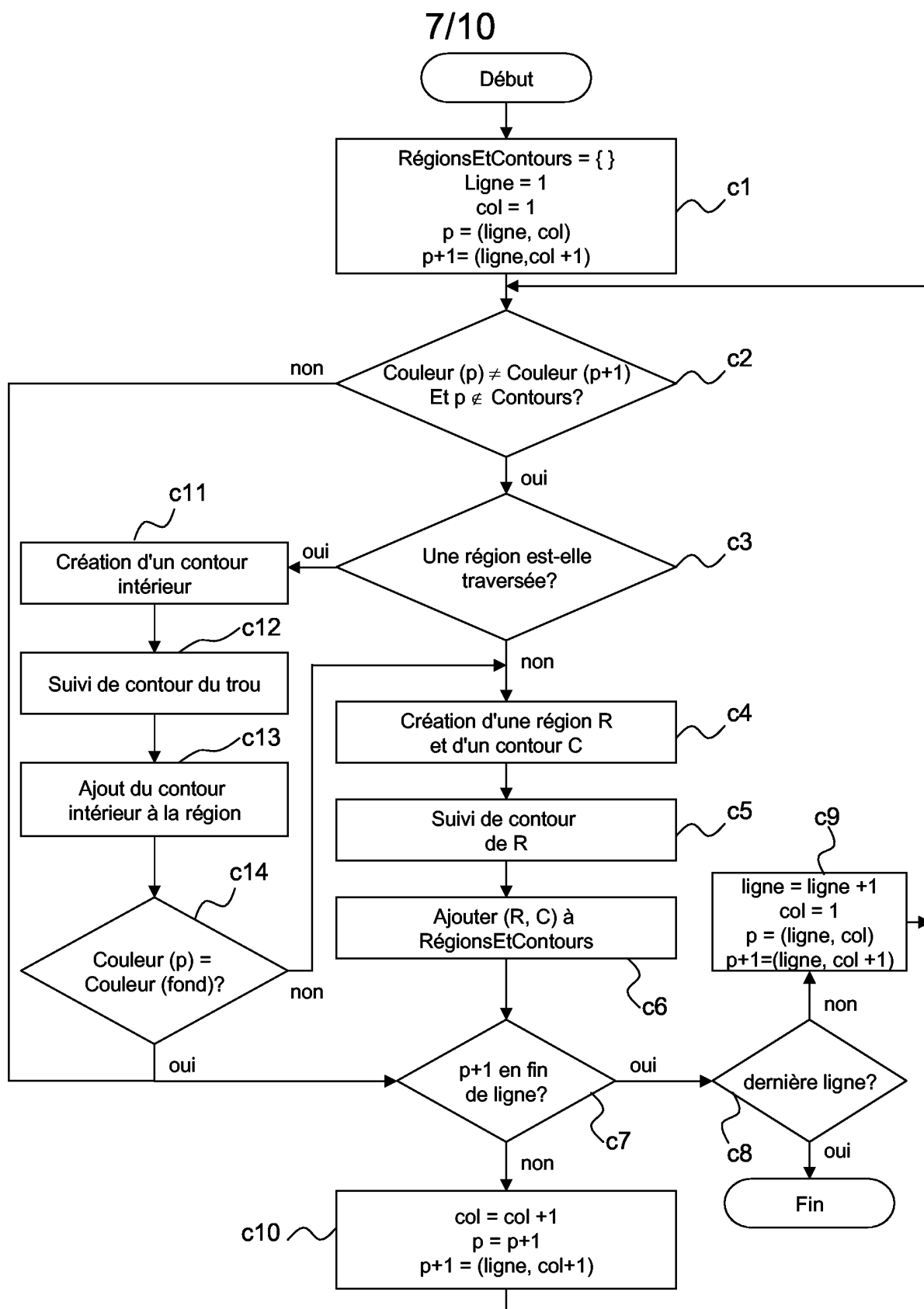


Fig. 10

8/10

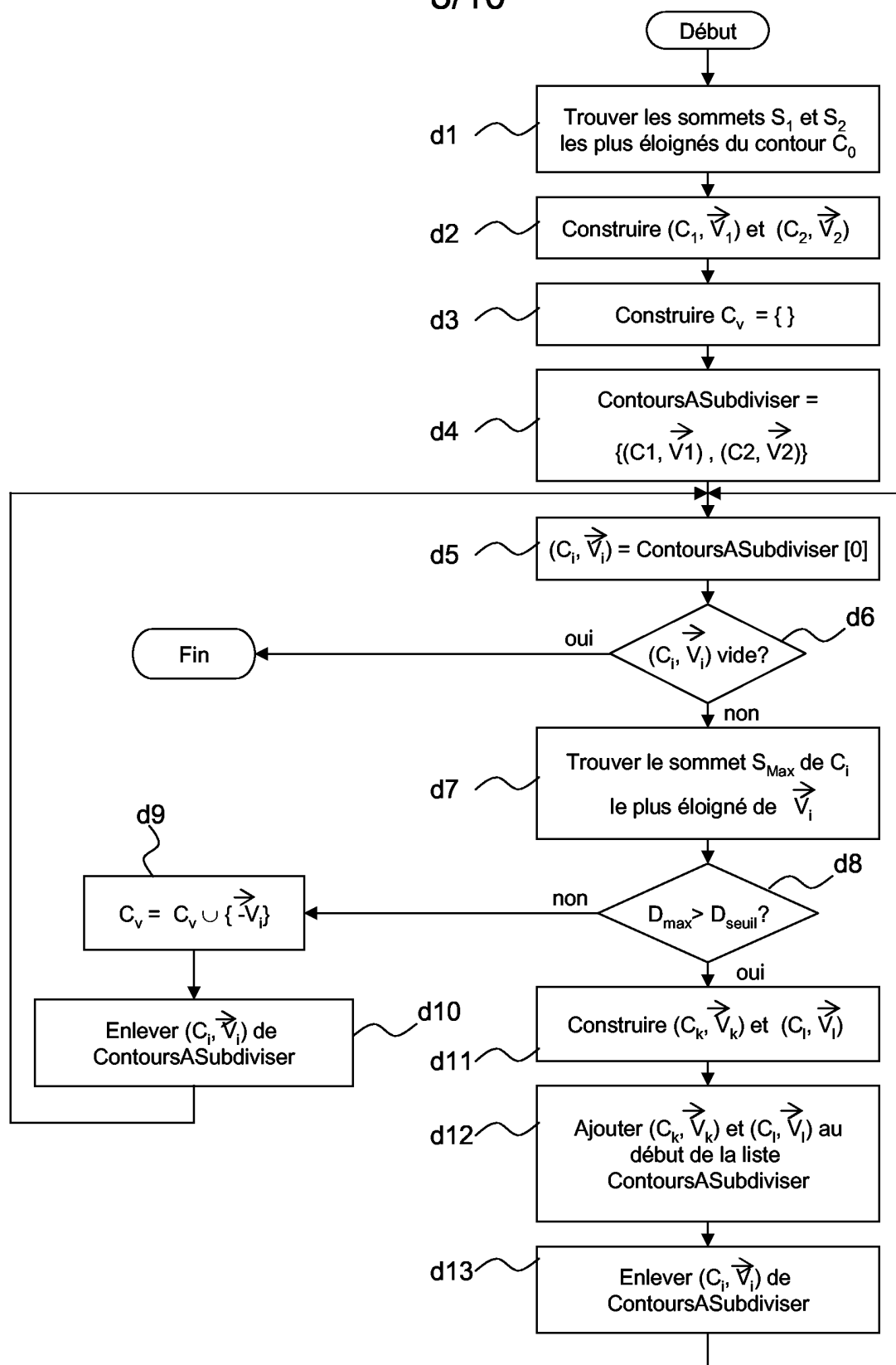
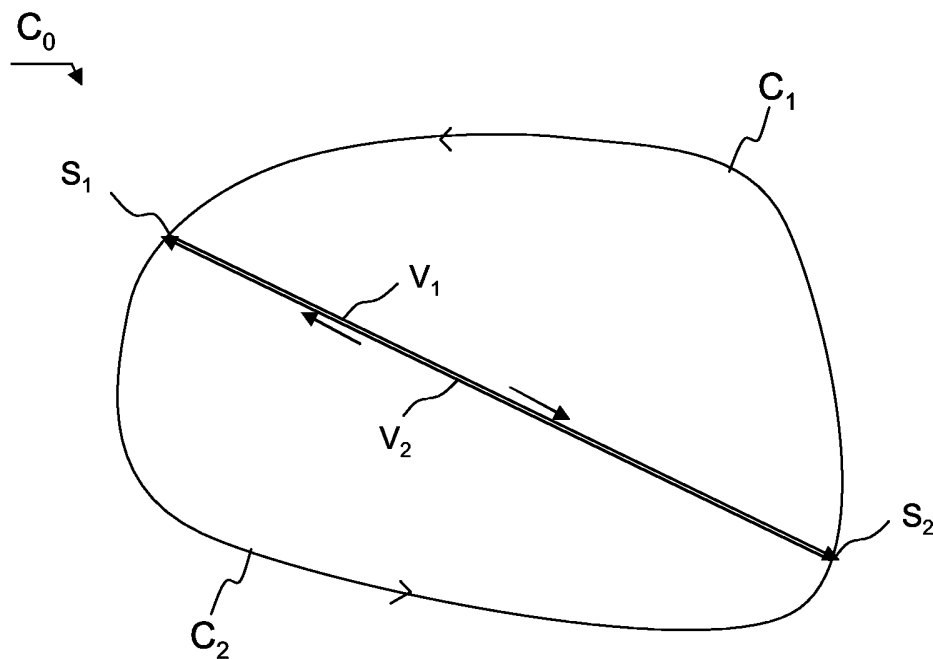
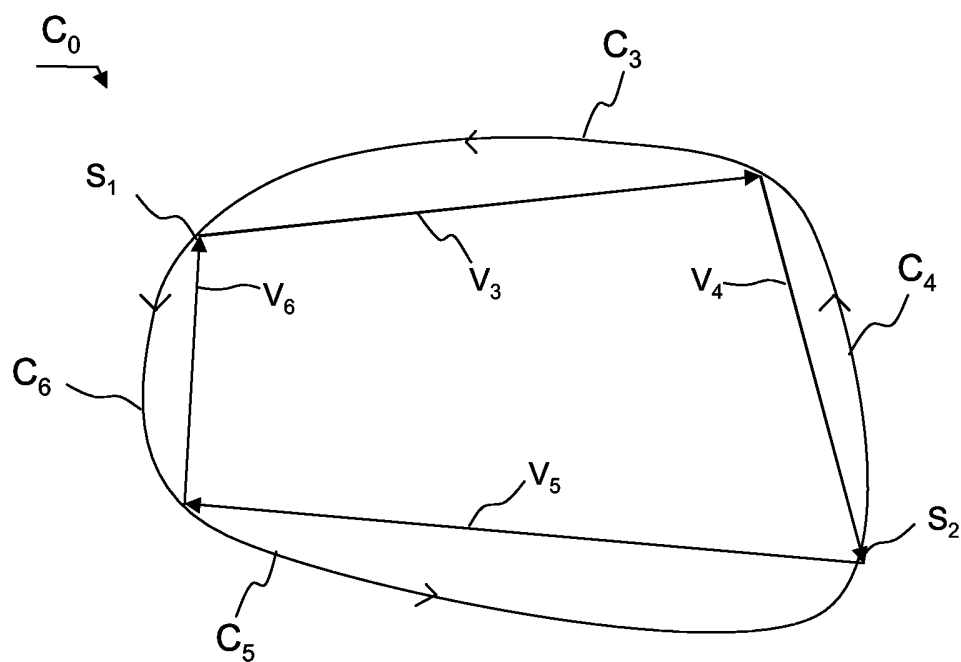


Fig. 11

9/10

Fig. 12Fig. 13

10/10

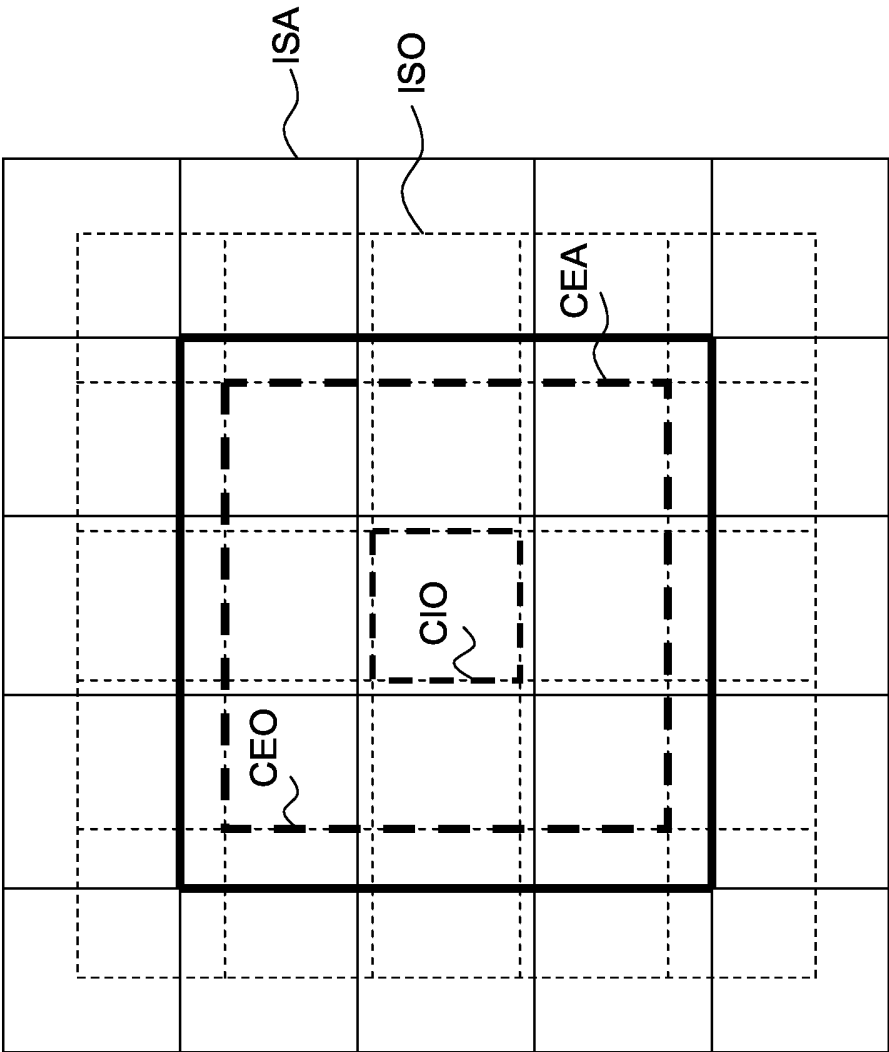


Fig. 14