

(19)



Europäisches Patentamt

European Patent Office

Office européen des brevets



(11)

EP 1 691 342 A1

(12)

DEMANDE DE BREVET EUROPEEN

(43) Date de publication:

16.08.2006 Bulletin 2006/33

(51) Int Cl.:

G09G 5/04 ^(2006.01)**G09G 3/20** ^(2006.01)(21) Numéro de dépôt: **06290242.4**(22) Date de dépôt: **13.02.2006**

(84) Etats contractants désignés:

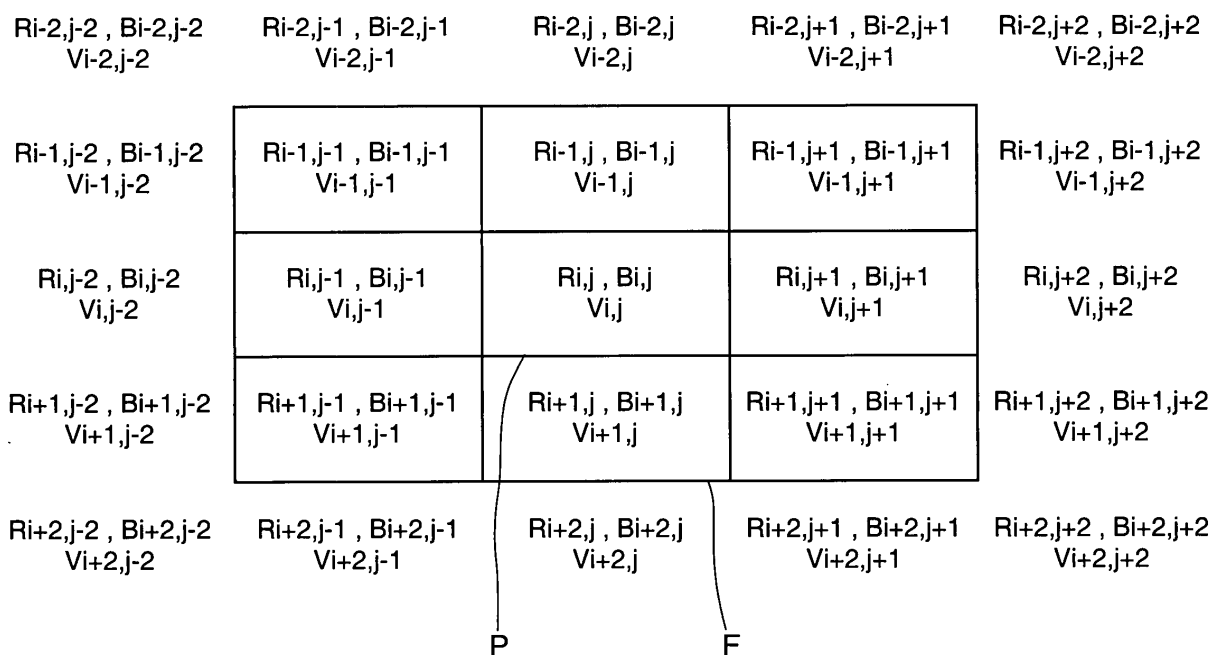
**AT BE BG CH CY CZ DE DK EE ES FI FR GB GR
HU IE IS IT LI LT LU LV MC NL PL PT RO SE SI
SK TR**

Etats d'extension désignés:

AL BA HR MK YU(30) Priorité: **14.02.2005 FR 0501469**(71) Demandeur: **STMicroelectronics SA**
92120 Montrouge (FR)(72) Inventeur: **Dumarest, Jacques**
38220 Notre Dame de Mesage (FR)(74) Mandataire: **Zapalowicz, Francis**
Bureau Casalonga & Josse
Bayerstrasse 71/73
80335 München (DE)(54) **Procédé et dispositif de traitement d'image**

(57) Ce procédé de traitement d'image pour la conversion d'une image d'un premier nombre de bits par pixel, par exemple 16 bits par pixel, en une image d'un deuxième nombre de bits par pixel, par exemple 18 bits

par pixel, consiste à élaborer des valeurs de bits additionnels de codage des composantes (R_i , V_i , B_i) de chaque pixel (i) à partir des valeurs de composantes de pixels adjacents et à élaborer le signal converti à partir des bits additionnels de codage.

FIG.5**EP 1 691 342 A1**

Description

[0001] L'invention concerne l'affichage ou l'impression d'images de haute qualité, et notamment, l'affichage d'images en mode dit « de couleurs réelles » ou « true color », en anglais, en particulier pour dispositif d'affichage à cristaux liquides LCD.

[0002] Une application particulièrement intéressante de l'invention concerne l'affichage d'images sur des écrans LCD de postes téléphoniques cellulaires ou assistant personnel (PDA) qui tendent, actuellement, à permettre d'afficher des images de haute qualité, c'est-à-dire des écrans qui sont capables d'afficher 262 144 couleurs simultanément.

[0003] Comme on le sait, la plupart des couleurs à afficher peuvent être reproduites à partir de trois couleurs primaires, à savoir le rouge, le vert et le bleu (RVB ou RGB pour « red, green, blue »). Aussi, les pixels, qui constituent les images à afficher sur les écrans ou à imprimer, sont associés chacun à des valeurs de composantes de rouge, de vert et de bleu. Ces valeurs de composantes sont codées, pour chaque pixels, sous la forme d'une trame de bits dont la longueur définit le nombre de couleurs maximum de l'image affichée.

[0004] C'est ainsi qu'une trame de 16 bits permet d'afficher 65 536 couleurs, qu'une trame de 18 bits permet d'afficher 262 144 couleurs, alors qu'une trame de 24 bits permet d'afficher 16 millions de couleurs simultanément.

[0005] Actuellement, les écrans à cristaux liquides à LCD utilisés dans les équipements portatifs tels qu'un téléphone cellulaire ou PDA ne permettent d'afficher que 262 144 couleurs, ce qui correspond à une trame de 18 bits. Une telle trame comprend 6 bits de codage de la composante de rouge, 6 bits de codage de la composante de vert et 6 bits de codage de la composante de bleu.

[0006] Ainsi, pour fonctionner de manière optimale, les images délivrées à un afficheur d'une capacité de 262 144 couleurs doivent être codées au moyen d'une trame de 18 bits par pixels.

[0007] Or, les algorithmes de traitement d'images et les équipements matériels, en particulier les mémoires de trame ne permettent généralement pas de supporter un tel format.

[0008] En effet, les mémoires de trame, qui servent à mémoriser les images avant leur affichage sur un écran, ne sont généralement adaptées que pour le stockage de trames de 16 bits par pixel, de 24 bits ou 32 bits par pixel, soit un multiple de 8 bits (Byte).

[0009] On a représenté sur les figures 1 et 2 l'organisation d'une telle mémoire de trame organisée en mots de 32 bits pour le stockage d'une trame de 16 bits par pixel et pour le stockage d'une trame de 24 bits par pixel, respectivement.

[0010] Comme on le voit sur la figure 1, dans un format de 16 bits par pixel, les composantes de bleu et de vert sont codées au moyen de 5 bits, alors que 6 bits servent au codage de la composante de rouge. Chaque adresse mémoire permet le stockage des valeurs de composantes de rouge, de vert et de bleu de 2 pixels (n et n+1) d'une image.

[0011] En ce qui concerne le stockage d'une trame de 24 bits par pixel, 8 bits servent au codage des composantes de rouge, de vert et de bleu. Quatre pixels sont stockés dans trois mots mémoire. Comme il est possible de stocker les bits de quatre pixels dans trois adresses mémoire de 32 bits ($4 \times 24 = 3 \times 32$), ceci a pour conséquence qu'un pixel peut se trouver « à cheval » sur deux mots mémoire (pixels n+1 et n+2), de sorte que si l'on souhaite modifier un pixel, deux mots doivent être traités entraînant un temps de traitement et une bande passante beaucoup plus élevés.

[0012] En se référant à la figure 3, lorsque l'on utilise une mémoire de trame de 24 bits par pixel, une qualité optimale est obtenue pour l'image affichée. Cependant, comme un panneau de cristaux liquides d'un équipement portatif ne supporte généralement au maximum que 18 bits par pixel, les deux bits les moins significatifs des mots de codage des composantes de couleurs ne sont pas connectés au panneau (troncature de la valeur exact)

[0013] Cette solution permet l'affichage de 262 144 couleurs mais au prix d'une augmentation de la taille de la mémoire de 50 % par rapport à une mémoire trame de 16 bits par pixel. En outre, la bande passante nécessaire est aussi augmentée de 50%.

[0014] En ce qui concerne la qualité des images obtenues, le rapport PSNR (« Peak Signal to Noise Ratio ») est de 42,69 décibels pour chaque composante de couleur de 8 bits tronquée à 6 bits.

[0015] Ce rapport est, de manière conventionnelle, calculé à partir d'un calcul de l'erreur quadratique moyenne MSE entre des valeurs de pixels de référence et des valeurs de pixels résultant de la troncature.

[0016] En se référant maintenant à la figure 4, sur laquelle on a représenté l'utilisation d'une mémoire de trame de 16 bits par pixel pour l'affichage d'images sur un panneau LCD de 18 bits par pixel, on voit que, dans ce cas, les mots de codage de 5 bits de rouge et de bleu et le mot de codage de la composante de vert, codé sous 6 bits, sont intégralement transmis dans des entrées correspondantes du panneau LCD. Toutefois, dans ce cas, les bits LSB les moins significatifs des composantes de rouge et de bleu de la mémoire de trame ne sont pas connectés. Dans l'exemple de réalisation représenté, les bits LSB des composantes de rouge et de bleu sont positionnés à zéro.

[0017] Les mots mémorisés de codage des composantes de rouge et de bleu peuvent toutefois être respectivement élaborés soit par troncature, soit par calcul d'une valeur arrondie à partir de la valeur 24 bits.

[0018] Comme indiqué précédemment, la troncature consiste à ne pas considérer les bits les moins significatifs des mots de codage des composantes de rouge et de bleu. Dans ce cas, le rapport PSNR pour les composantes de rouge et de bleu est de 35,70 décibels et le rapport PSNR pour la composante de vert est de 42,69 décibels.

[0019] En ce qui concerne le calcul de la valeur arrondie, celle-ci présente un rapport PSNR de 40,72 décibels en ce qui concerne les composantes de rouge et de bleu et un rapport PSNR de 46,37 décibels en ce qui concerne la composante de vert.

[0020] Au vu de ce qui précède, le but de l'invention est de pallier les inconvénients des systèmes d'affichage de l'état de la technique et, de manière générale, de permettre l'affichage d'une image de haute qualité, en particulier une image en 262 144 couleurs en utilisant une mémoire de trame de capacité réduite.

[0021] Plus particulièrement, un autre but de l'invention est de permettre d'afficher une image de 262 144 couleurs à partir d'une image de 16 bits par pixels extrait d'une mémoire de trame.

[0022] Selon un premier aspect, l'invention a donc pour objet un procédé de traitement d'image, pour la conversion d'un signal d'un premier nombre de bits par pixel en un signal d'un deuxième nombre de bits par pixel supérieur audit premier nombre de bits par pixel, le signal à convertir et le signal converti comprenant un ensemble de pixels associés chacun à des valeurs de composantes de rouge, de vert et de bleu constituant une trame dudit premier nombre de bits et une trame dudit deuxième nombre de bits, respectivement.

[0023] Selon ce procédé, on élabore des valeurs de bits additionnels de codage des composantes de chaque pixel à partir des valeurs de composantes de pixels adjacents, les bits additionnels de codage étant utilisés pour l'élaboration du signal converti.

[0024] Selon une autre caractéristique de l'invention, le signal à convertir et le signal converti comprennent respectivement un ensemble de pixels associés à des valeurs de composantes de rouge, de vert et de bleu constituant une trame de 16 bits par pixel et une trame de 18 bits par pixel.

[0025] Dans un mode de mise en oeuvre, l'image à convertir étant constituée d'une matrice de ligne et de colonne de pixels, on élabore les valeurs des bits additionnels en considérant les pixels d'une fenêtre d'analyse de 3 x 3 pixels.

[0026] En variante, on élabore les valeurs des bits additionnels en considérant les pixels d'une fenêtre d'analyse de 5 x 5 pixels.

[0027] On pourrait également utiliser des fenêtres d'analyse d'un nombre quelconque de pixels, de préférence un nombre impair de pixels, englobant le pixel à traiter.

[0028] Pour procéder à l'élaboration du signal converti, on affecte la valeur des bits de codage de la composante verte de l'image à convertir aux bits de codage de la composante verte de l'image convertie, la composante verte étant déjà codée sur 6 bits.

[0029] Au contraire, on calcule la valeur des bits de codage de la composante rouge et de la composante bleue par calcul, pour chacune desdites composantes rouge et bleue, de la différence entre les valeurs des composantes rouge et verte, d'une part, et de la différence entre les valeurs des composantes bleue et verte, d'autre part, par calcul d'une valeur filtrée à partir des valeurs de différences calculées et par affectation d'un bit additionnel aux valeurs des composantes rouge et bleue de l'image à convertir en fonction de la valeur filtrée.

[0030] Pour procéder au filtrage, on applique des coefficients de pondération aux valeurs de différence calculées.

[0031] On additionne en outre les valeurs pondérées et on divise le résultat de ladite addition par la somme des coefficients de pondération de la fenêtre d'analyse.

[0032] Ainsi, dans le cas où la différence entre la valeur filtrée de la composante rouge, d'une part et de la composante bleue, d'autre part et la valeur non filtrée desdites composantes, respectivement est supérieure à 2, le bit additionnel est positionné à 1.

[0033] Dans le cas contraire, dans le cas où ladite différence est inférieure ou égale à - 2, le bit additionnel est obtenu en soustrayant le bit 1 à l'endroit du bit additionnel.

[0034] Sinon, c'est-à-dire si la valeur de ladite différence est inférieure ou égale à 2 ou supérieure à - 2, la valeur de la composante rouge ou de la composante bleue de l'image à convertir est affectée à la valeur de la composante rouge ou de la composante bleue de l'image convertie (valeur inchangée).

[0035] Selon un autre mode de mise en oeuvre de l'invention, on compare la valeur d'un pixel en cours de traitement et de pixels adjacents avec une valeur de seuil et on affecte au pixel en cours de traitement une valeur maximale si ce pixel et les pixels adjacents ont tous une valeur supérieure à la valeur de seuil.

[0036] Selon un deuxième aspect, l'invention a pour objet un dispositif de traitement d'image pour la conversion d'un signal d'un premier nombre de bits par pixel en un signal d'un deuxième nombre de bits par pixel supérieur au premier nombre de bits par pixel, le signal à convertir et le signal converti comprenant un ensemble de pixels associés chacun à des valeurs de composantes de rouge, de vert et de bleu constituant une trame dudit premier nombre de bits et une trame dudit deuxième nombre de bits, respectivement, caractérisé en ce qu'il comporte des moyens pour élaborer des valeurs de bits additionnels de codage des composantes de chaque pixel à partir des valeurs de composantes de pixels adjacents, et des moyens pour élaborer le signal converti à partir des bits de codage des valeurs des composantes de rouge, de vert et de bleu du signal à convertir et des bits additionnels de codage.

[0037] L'invention a également pour objet un écran plat, comprenant un ensemble de pixels commandés par un ou plusieurs circuits de commande périphérique convertissant un signal d'un premier nombre de bits par pixel en un signal d'un deuxième nombre de bits par pixel supérieur au premier nombre de bits par pixel, le signal à convertir et le signal

converti comprenant un ensemble de pixels associés chacun à des valeurs de composantes de rouge, de vert et de bleu constituant une trame dudit premier nombre de bits et une trame dudit deuxième nombre de bits, respectivement, caractérisé en ce qu'il comporte des moyens pour élaborer des valeurs de bits additionnels de codage des composantes de chaque pixel à partir des valeurs de composantes de pixels adjacents et des moyens pour élaborer le signal converti à partir des bits de codage des valeurs des composantes de rouge, de vert et de bleu du signal à convertir et desdits bits additionnels.

[0038] Selon un mode de réalisation particulier de l'invention, l'écran plat peut être un écran LCD, TFT, OLED ou CRT.

[0039] Selon encore un autre aspect, l'invention a pour objet un équipement électronique portable, du type téléphone mobile, caractérisé en ce qu'il comporte un écran plat tel que défini ci-dessus.

[0040] Par équipement électronique, on entend, dans le cadre de la présente description, tout type de poste ou terminal portable ou mobile tel qu'un téléphone mobile, un PDA, un appareil de prise de vues ou tout autre dispositif de ce type pourvu d'un écran plat.

[0041] D'autres buts, caractéristiques et avantages de l'invention apparaîtront à la lecture de la description suivante, donnée uniquement à titre d'exemple non limitatif, et faite en référence aux dessins annexés sur lesquels :

- les figures 1 et 2, dont il a déjà été fait mention, illustrent l'architecture d'une mémoire de trame de 16 bits par pixel et d'une mémoire de trame de 24 bits par pixel, respectivement ;
- la figure 3 et la figure 4 précédemment évoquées illustrent respectivement l'affichage d'une image de 18 bits par pixel sur un panneau LCD à partir d'une mémoire de trame de 24 bits par pixel et d'une mémoire de trame de 16 bits par pixel, respectivement ;
- la figure 5 est un schéma illustrant le principe de conversion selon l'invention ;
- la figure 6 est un organigramme illustrant les principales phases d'un procédé de traitement d'image conforme à l'invention ;
- les figures 7 et 8 sont d'autres schémas illustrant le principe de conversion mis en oeuvre par le procédé de traitement d'image conforme à l'invention ;
- la figure 9 est un tableau comparatif illustrant les avantages procurés par le procédé de conversion conforme à l'invention sur l'image obtenue ; et
- la figure 10 montre un exemple de réalisation d'un dispositif de traitement d'image mettant en oeuvre le procédé de conversion selon l'invention.

[0042] Dans la suite de la description, on va décrire un exemple de mise en oeuvre d'un procédé de traitement d'image conforme à l'invention.

[0043] Dans l'exemple de réalisation considéré, ce procédé est destiné à permettre d'afficher une image de haute qualité avec 262 144 couleurs sur un dispositif d'affichage LCD de 18 bits par pixel (18 bpp) tout en utilisant une mémoire de trame de 16 bits par pixel (16bpp), et en conservant la bande passante requise.

[0044] Cependant, le principe selon l'invention pourrait également être généralisé pour la conversion d'autres formats de données.

[0045] De même, l'invention s'applique aussi bien à un traitement d'image pour l'affichage de l'image sur un écran que pour l'impression de cette image.

[0046] Comme indiqué précédemment, dans l'application envisagée, l'image est constituée d'un ensemble de pixels associés chacun à des valeurs de composante de rouge, de vert et de bleu codées, pour chaque pixel, au moyen d'une trame de 16 bits, à savoir 6 bits pour le vert, 5 bits pour le rouge et 5 bits pour le bleu.

[0047] Au contraire, pour l'affichage de 262 144 couleurs, les pixels sont chacun associés à des valeurs de composantes de rouge, de vert et de bleu constituant une trame de 18 bits, à savoir 6 bits pour le vert, le rouge et le bleu.

[0048] Il convient donc, selon l'invention, de convertir l'image de 16 bpp en une image de 18 bpp.

[0049] Il convient également de rappeler que l'image incidente est issue d'un capteur de type CCD ou CMOS pourvu d'un filtre couleur dit « filtre de Bayer ». Cette technologie d'acquisition d'images est largement répandue et ne sera donc pas décrite en détail par la suite.

[0050] On notera cependant que, selon cette technique, l'acquisition de l'image s'effectue au moyen de filtres de trois couleurs, à savoir rouges, verts et bleus, placés selon un réseau bidimensionnel de sorte que chaque pixel détecté corresponde à la couleur du filtre de ce pixel. Les filtres sont configurés de sorte que 50 % des filtres soient verts, que 25 % des filtres soient rouges et que 25 % des filtres soient bleus.

[0051] Ainsi, dans la mosaïque obtenue, la moitié des pixels sont verts, un quart des pixels sont rouges et un quart des pixels sont bleus, ce qui correspond au maximum de sensibilité de l'oeil humain. Pour chaque pixel, il convient alors de récupérer les composantes des couleurs manquantes. Ceci est généralement effectué grâce à des méthodes d'interpolation conventionnelles, en utilisant les valeurs des composantes correspondantes des pixels adjacents.

[0052] Comme on le sait, selon cette technique, il existe une corrélation spatiale entre les composantes rouge et bleue mais également une corrélation entre les composantes rouge et bleue et la composante verte.

[0053] Dans le cas où la source d'image est non pas un capteur CMOS ou CCD, mais une image compressée aux normes JPEG ou MPEG, le format d'origine est usuellement codé en information de luminance et de chrominance bleue et rouge : Y, Cb, Cr avec un sous échantillonnage des chrominances du type 4:2:2 ou 4:2:0 qui signifie qu'il y a moitié ou quart d'information de chrominance par rapport à la luminance.

[0054] En outre, dans le format 16bpp, dans la mesure où 6 bits sont utilisés pour le codage de la composante verte alors que 5 bits sont utilisés pour le codage de chacune des composantes rouge et bleue, la définition est supérieure en ce qui concerne la composante verte.

[0055] Au vu de ce qui précède, le procédé de conversion selon l'invention utilise la corrélation spatiale qui existe entre les composantes rouge et bleue et la corrélation qui existe entre les composantes rouge et bleue, d'une part, et la composante verte d'autre part, et est également basée sur la définition supérieure de la composante verte.

[0056] Comme indiqué précédemment, il convient d'élaborer une trame de 18 bpp, c'est-à-dire une trame selon laquelle les composantes de rouge, de vert et de bleu sont chacune codées par un mot de 6 bits, à partir d'une trame de 16 bpp issu de la mémoire de trame dans laquelle, pour chaque pixel, les composantes rouge et bleue sont chacune codées au moyen d'un mot de 5 bits, alors que la composante verte est codée au moyen d'un mot de 6 bits.

[0057] Aussi, en ce qui concerne la composante verte, le mot de 6 bits de la composante verte de chaque pixel à convertir est recopié dans un mot correspondant de codage de la composante verte de chaque pixel converti.

[0058] En ce qui concerne les composantes rouge et bleue, il convient, pour chaque pixel, d'élaborer un bit additionnel qui doit être ajouté aux 5 bits de codage des composantes rouge et bleue de chaque pixel du signal à convertir. L'élaboration de ces bits additionnels est basée sur la corrélation existante entre les composantes rouge, verte et bleue.

Aussi, en se référant à la figure 5, on utilise une fenêtre d'analyse qui prend en compte les pixels voisins d'un pixel P en cours de traitement. Comme on le voit sur cette figure 5, les pixels se présentent sous la forme d'une matrice de ligne i et de colonne j de pixels associés chacun à des composantes RVB de rouge, de vert et de bleu. Pour l'élaboration des bits manquants des composantes de rouge et de bleu d'un pixel P en cours d'analyse, on prend en compte les pixels immédiatement voisins du pixel P.

[0059] Dans l'exemple de réalisation représenté sur la figure 5, la fenêtre d'analyse est une fenêtre de 3 x 3 pixels. On pourrait cependant, en variante, utiliser une fenêtre de cinq pixels ou tout autre fenêtre de dimension appropriée pour l'utilisation envisagée, d'un nombre impair de pixels.

[0060] Ainsi, en se référant à la figure 6, au cours d'une première étape 10, on procède à l'acquisition des valeurs de composantes rouge, verte et bleue de l'ensemble des 9 pixels de la fenêtre d'analyse F.

[0061] En se référant également à la figure 7, sur laquelle seuls les pixels détectés par la fenêtre d'analyse F ont été considérés, et qui correspondent à une étape de traitement correspondant à l'élaboration d'un bit d'un mot de codage de la composante rouge, lors de l'étape 12 suivante, il est procédé à un calcul de la différence entre les valeurs des composantes rouge et verte $R_{ij} - V_{ij}$ pour chacune des 9 valeurs de la fenêtre d'analyse de trois fois trois pixels.

[0062] Les valeurs de différence ainsi calculées sont, lors de l'étape 14 suivante, filtrées en utilisant la fenêtre de filtrage représentée à la figure 8. Ce filtrage consiste à multiplier chaque valeur de différence calculée lors de l'étape 12 précédemment par un coefficient de pondération C_{ij} qui dépend de la position du pixel considérée, en affectant un coefficient de pondération maximum à la différence entre les composantes rouge et verte du pixel P en cours d'analyse, par exemple égal à 4.

[0063] Lors de l'étape 16 suivante, la phase de filtrage se poursuit en additionnant les valeurs filtrées ainsi obtenues et en divisant le résultat de l'addition par la somme des coefficients de pondération de la fenêtre.

[0064] Lors de l'étape 18 suivante, on procède à un calcul de la différence entre la valeur filtrée ainsi obtenue et la valeur non filtrée.

[0065] Il s'agit en l'occurrence de soustraire la valeur de la différence entre les composantes rouge et verte du pixel P en cours d'analyse avec la valeur de la différence filtrée élaborée, comme indiqué précédemment, en utilisant les valeurs de composante des pixels voisins.

[0066] Lors des étapes 20 et 22 suivantes, le résultat de la comparaison ainsi effectuée est comparé avec des valeurs de seuil prédéterminées.

[0067] Par exemple, dans l'exemple de réalisation considéré, si la différence calculée entre les valeurs filtrées et non filtrées est supérieure à 2, alors on affecte la valeur 1 au bit de codage additionnel (étape 22).

[0068] En d'autres termes, en considérant que les cinq bits de codage de la composante rouge de chaque pixel à convertir sont constitués par les cinq bits les plus significatifs d'un octet xxxxx000 de codage de cette composante rouge (les signes x désignant des valeurs utiles du mot de codage) au cours de l'étape 22, on ajoute à ce mot la valeur binaire 100. On obtient donc le mot de codage converti suivant : xxxxx100.

[0069] Au contraire, si lors de l'étape 20 précédente, il a été détecté que la différence entre la valeur filtrée et la valeur non filtrée est inférieure ou égale à deux, lors de l'étape 24 suivante, on détecte si cette différence est inférieure ou égale à - 2. Si tel est le cas, lors de l'étape 26 suivante, on soustrait au mot de codage de la composante rouge la valeur binaire 100. On obtient alors le mot de codage suivant : yyyyy100.

[0070] Enfin, dans le cas contraire, c'est-à-dire si la différence entre la valeur filtrée et la valeur non filtrée est supérieure

à - 2, alors le mot de codage est maintenu inchangé. En d'autres termes, le troisième bit est positionné à 0. Le mot de codage de la composante rouge du pixel converti est alors : xxxxx000.

[0071] Le traitement décrit ci-dessus est, de même, effectué pour la composante bleue. En d'autres termes, après acquisition des valeurs des composantes rouge, bleue et verte de chaque pixel (étape 10), la différence entre les composantes bleue et verte est calculée pour chacune des neuf valeurs de la fenêtre de trois fois trois pixels. La différence ainsi calculée est alors filtrée, comme mentionné précédemment. La valeur filtrée est alors comparée avec les valeurs de seuil précédemment indiquées afin soit d'ajouter ou de soustraire le mot binaire 100, en fonction de la position du bit de codage additionnel, ou de conserver inchangé le troisième bit du mot de codage de la composante bleue.

[0072] L'algorithme utilisé pour procéder à la conversion décrite ci-dessus d'une image de 16 bpp en une image de 18 bpp peut par exemple être réalisé à partir des lignes de code suivantes :

```

for (i=0 ; i<Height ; i++)
{
    if ((i !=0 && (i !=Height-1)))
        for j=0 ; j<Width*3 ; j=j+3)
        {
            if ((j !=0 && (j !=Width*3-3)))
            {
                // Red Processing

                // Compute Delta between Red and Green Value for the 9 pixels
                Delta[0]=Buffer1[((i-1)*Width*3)+j-3]-Buffer1[((i-1)*Width*3)+j-2]
                Delta[1]=Buffer1[((i-1)*Width*3)+j]-Buffer1[((i-1)*Width*3)+j-1]
                Delta[2]=Buffer1[((i-1)*Width*3)+j+3]-Buffer1[((i-1)*Width*3)+j+4]
                Delta[3]=Buffer1[(i*Width*3)+j-3]-Buffer1[(i*Width*3)+j-2]
                Delta[4]=Buffer1[(i*Width*3)+j]-Buffer1[(i*Width*3)+j+1]
                Delta[5]=Buffer1[(i*Width*3)+j+3]-Buffer1[(i*Width*3)+j+4]
                Delta[6]=Buffer1[((i+1)*Width*3)+j-3]-Buffer1[((i+1)*Width*3)+j-2]
                Delta[7]=Buffer1[((i+1)*Width*3)+j]-Buffer1[((i+1)*Width*3)+j+1]
                Delta[8]=Buffer1[((i+1)*Width*3)+j+3]-Buffer1[((i+1)*Width*3)+j+4]
                Filtered_Delta=(Delta[0]+2*Delta[1]+Delta[2]+2*Delta[3]+4*Delta[4]
                    +2*Delta[5]+Delta[6]+2*Delta[7]+Delta[8])/16
                Current_Red=Buffer1[(i*Width*3)+j]
                Current_Green=Buffer1[(i*Width*3)+j+1]
            }
        }
    }

```

```

    if ((Filtered_Delta-Delta[4]>2)
5      Current_Red=Current_Red+4
    Else if ((Filtered_Delta-Delta[4])<=-2)
    {
        If (Current_Red !=0)
10      Current_Red=Current_Red-4
    }

    Buffer2[(i*Width*3)+j]=Current_Red
15
}
else
{
20      Buffer2[(i*Width*3)+j]=Buffer1[(i*Width*3)+j]
    }
}
}
25
else
{
30      for(j=0 ; j<Width*3 ; j=j+3)
    {
        Buffer2[(i*Width*3)+j]=Buffer1[(i*Width*3)+j]
        Buffer2[(i*Width*3)+j+1]=Buffer1[(i*Width*3)+j+1]
35      Buffer2[(i*Width*3)+j+2]=Buffer1[(i*Width*3)+j+2]
    }
}
}
40

```

[0073] On notera que cet algorithme ne constitue qu'un exemple non optimisé de mise en oeuvre de l'invention permettant d'illustrer les traitements effectués. Il décrit uniquement le traitement sur la composante rouge, le traitement pour la composante bleu étant identique. Les valeurs rouge, vert et bleu sont codées sur des octets : 8 bits.

[0074] En se référant à la figure 9, sur laquelle on a présenté diverses valeurs de rapport PSNR, en ce qui concerne notamment les composantes rouge, verte et bleue et la taille T de la mémoire de trame pour une mémoire de trame de 24 bits par pixel, avec deux bits LSB non connectés ; pour une mémoire de trame de 16 bits par pixel générée à partir de la valeur originelle 24bpp par troncature ou arrondi et pour une mémoire de 16 bpp générée à partir de la valeur originelle 24bpp par arrondi puis affichée en 18bpp en utilisant une conversion conformément à l'invention, on voit que la conversion effectuée selon le procédé conforme à l'invention permet un gain moyen de deux décibels par rapport à l'utilisation d'une mémoire de trame de 24 bits par pixel et un gain de 1 ou de 5 décibels par rapport à l'utilisation d'une mémoire de trame de 16 bits par pixel.

[0075] On notera que, sur la figure 9, les valeurs de PSNR sont des valeurs calculées, sauf pour la dernière colonne, qui sont le résultat de mesures de la qualité de l'algorithme présenté ci-dessus et qui dépendent du contenu de l'image.

[0076] On notera enfin, en se référant maintenant à la figure 10 que le procédé de conversion conforme à l'invention peut être incorporé à un dispositif 30 réalisé par exemple sous la forme d'une interface interposée entre une mémoire de trame 32 de 16 bits par pixel et un panneau 34 d'affichage LCD de 18 bits par pixel pour l'affichage d'images de 262

144 couleurs.

[0077] L'interface comporte alors des moyens logiciels et matériels de calcul dûment programmés pour élaborer des valeurs de bits additionnels de codage, et pour élaborer l'image convertie à partir des bits additionnels, selon le procédé décrit précédemment.

[0078] Bien entendu, l'invention n'est pas limitée au mode de réalisation décrit.

[0079] Dans la description du mode de réalisation de l'invention qui vient d'être faite, on ajoute, dans une trame de bits de codage de chaque composante de bleu et de rouge, un bit de codage supplémentaire en positionnant à « 1 » ou « 0 » un bit additionnel de la trame ou, comme indiqué précédemment, en maintenant inchangée la valeur de la composante rouge ou bleue d'une image à convertir.

[0080] La valeur du bit de codage supplémentaire est déterminée en fonction des pixels adjacents, relativement à un pixel considéré et, en particulier, par calcul de la différence entre les composantes de rouge et de vert, d'une part, et des composantes de bleu et de vert, d'autre part, et par calcul d'une valeur moyenne pondérée sur un ensemble de pixels adjacents et élaboration du bit additionnel en fonction de la valeur moyenne pondérée.

[0081] Afin de s'assurer que l'ensemble des couleurs peuvent être obtenues, dans un autre mode de réalisation, on contrôle la valeur de chaque pixel et de ses pixels adjacents et on lui attribue une valeur maximale si la valeur de tous les pixels dépasse une valeur de seuil.

[0082] Ainsi, par exemple, en se référant à la figure 8, on considère les quatre pixels voisins d'un pixel en cours de traitement qui sont affectés d'un coefficient « 2 ». Il s'agit ainsi de considérer les pixels de gauche, de droite, et les pixels situés en haut et en bas. Si les cinq pixels ont tous une valeur initiale supérieure à la valeur de seuil, on affecte au pixel en cours de traitement la valeur maximale « 0xFC ».

[0083] La valeur de seuil est par exemple égale à « 0xF8 », ce qui traduit que les cinq bits de codage de la composante de rouge ou de bleu sont positionnés à « 1 ».

[0084] La valeur maximale affectée au pixel traité correspond à une valeur selon laquelle les six bits de codage sont positionnés à « 1 ».

[0085] On notera cependant que selon la présente description, l'ajout d'un bit supplémentaire de codage d'une composante de rouge ou de bleu consiste à positionner à « 0 » ou à « 1 » le troisième bit du mot de codage, considéré à partir du bit le moins significatif.

[0086] Il est également possible, en variante, de positionner à « 1 » ou « 0 » le troisième bit, considéré à partir du bit le plus significatif.

[0087] Dans ce cas, la valeur de seuil à laquelle sont comparées la valeur d'un pixel en cours de traitement et la valeur des quatre pixels adjacents devient « 0x1F », alors que la valeur maximale affectée au pixel analysé devient « 0x3F ».

[0088] Dans ce cas, l'algorithme utilisé pour procéder à la conversion d'une image de 16 bpp en une image de 18 bpp peut être réalisé à partir des lignes de code suivantes :


```

for (i=0 ; i<Height ; i++)
{
    if ((i !=0 && (i !=Height-1)))
        for j=0 ; j<Width*3 ; j=j+3)
            {
                if ((j !=0 && (j !=Width*3-3)))
                    {
// Red Processing

// Compute Delta between Red and Green Value for the 9 pixels
Delta[0]=Buffer1[((i-1)*Width*3)+j-3]-Buffer1[((i-1)*Width*3)+j-2]
Delta[1]=Buffer1[((i-1)*Width*3)+j]-Buffer1[((i-1)*Width*3)+j-1]
Delta[2]=Buffer1[((i-1)*Width*3)+j+3]-Buffer1[((i-1)*Width*3)+j+4]
Delta[3]=Buffer1[(i*Width*3)+j-3]-Buffer1[(i*Width*3)+j-2]
Delta[4]=Buffer1[(i*Width*3)+j]-Buffer1[(i*Width*3)+j+1]
Delta[5]=Buffer1[(i*Width*3)+j+3]-Buffer1[(i*Width*3)+j+4]
Delta[6]=Buffer1[((i+1)*Width*3)+j-3]-Buffer1[((i+1)*Width*3)+j-2]
Delta[7]=Buffer1[((i+1)*Width*3)+j]-Buffer1[((i+1)*Width*3)+j+1]
Delta[8]=Buffer1[((i+1)*Width*3)+j+3]-Buffer1[((i+1)*Width*3)+j+4]
Filtered_Delta=(Delta[0]+2*Delta[1]+Delta[2]+2*Delta[3]+4*Delta[4]
                +2*Delta[5]+Delta[6]+2*Delta[7]+Delta[8])/16
Current_Red=Buffer1[(i*Width*3)+j]
Current_Green=Buffer1[i*Width*3)+j+1]

if ((Filtered_Delta-Delta[4]>2)
    Current_Red=Current_Red+4
Else if ((Filtered_Delta-Delta[4])<=-2)
{
    If (Current_Red !=0)
        Current_Red=Current_Red-4

```

```

    }
    if (((Buffer1[((i-1)*Width*3)+j]) & 0xF8) == 0xF8) &&
5      ((Buffer1[(i*Width*3)+j-3]) & 0xF8) == 0xF8) &&
      (((Buffer1[(i*Width*3)+j]) & 0xF8) == 0xF8) &&
      (((Buffer1[(i*Width*3)+j+3]) & 0xF8) == 0xF8) &&
10     (((Buffer1[((i+1)*Width*3)+j]) & 0xF8) == 0xF8))
      Current_Red = 0xFC;
      Buffer2[(i*Width*3)+j]=Current_Red
    }
15   else
    {
      Buffer2[(i*Width*3)+j]=Buffer1[(i*Width*3)+j]
20     }
    }
  }
25   else
    {
      for(j=0 ; j<Width*3 ; j=j+3)
30     {
      Buffer2[(i*Width*3)+j]=Buffer1[(i*Width*3)+j]
      Buffer2[(i*Width*3)+j+1]=Buffer1[(i*Width*3)+j+1]
      Buffer2[(i*Width*3)+j+2]=Buffer1[(i*Width*3)+j+2]
35     }
    }
  }
40

```

Revendications

- 45 1. Procédé de traitement d'image pour la conversion d'une image d'un premier nombre de bits par pixel en une image d'un deuxième nombre de bits par pixel supérieur au premier nombre de bits par pixel, le signal à convertir et le signal converti comprenant respectivement un ensemble de pixels associés chacun à des valeurs de composantes (RVB) de rouge, de vert et de bleu constituant une trame dudit premier nombre de pixels et une trame dudit deuxième nombre de pixels, **caractérisé en ce que** l'on élabore des valeurs de bits additionnels de codage des composantes
50 de chaque pixel à partir des valeurs de composantes de pixels adjacents, les bits additionnels étant utilisés pour l'élaboration du signal converti.
2. Procédé selon la revendication 1, **caractérisé en ce que** l'image à convertir et l'image convertie comprennent respectivement un ensemble de pixels associés à des valeurs de composantes ($R_{i,j}$, $V_{i,j}$, $B_{i,j}$) de rouge, de vert et
55 de bleu constituant une trame de 16 bits par pixel et une trame de 18 bits par pixel, respectivement.
3. Procédé selon l'une des revendications 1 et 2, **caractérisé en ce que** l'image à convertir étant constituée d'une matrice de lignes et de colonnes de pixels, on élabore les valeurs des bits additionnels en considérant les pixels

d'une fenêtre (F) d'analyse de 3 X 3 pixels.

4. Procédé selon l'une des revendications 2 et 3, **caractérisé en ce que** l'image à convertir étant constituée d'une matrice de lignes et de colonnes de pixels, on élabore les valeurs des bits additionnels en considérant les pixels d'une fenêtre (F) d'analyse de 5 X 5 pixels.

5. Procédé selon l'une des revendications 3 et 4, **caractérisé en ce que** l'on affecte les valeurs des bits de codage de la composante verte de l'image à convertir aux bits de la composante verte de l'image convertie.

6. Procédé selon la revendication 5, **caractérisé en ce que** l'on calcule la valeur des bits additionnels de codage de la composante rouge et de la composante bleue par calcul, par chacune desdites composantes rouge et bleue, de la différence $(R_{i,j} - V_{i,j})$ entre les valeurs des composantes rouge et verte, d'une part et de la différence $(B_{i,j} - V_{i,j})$ entre les valeurs des composantes bleue et verte, d'autre part, par calcul d'une valeur filtrée à partir des valeurs de différence calculées et affectation d'un bit additionnel aux valeurs des composantes rouge et bleue de l'image à convertir en fonction de la valeur filtrée.

7. Procédé selon la revendication 6, **caractérisé en ce que** le filtrage est réalisé en appliquant des coefficients de pondération $(C_{i,j})$ aux valeurs de différence calculées.

8. Procédé selon la revendication 7, **caractérisé en ce que** l'on additionne les valeurs pondérées et l'on divise le résultat de ladite addition par la somme des coefficients de pondération de la fenêtre d'analyse.

9. Procédé selon l'une quelconque des revendications 6 à 8, **caractérisé en ce que** si la différence entre la valeur filtrée de la composante rouge, d'une part, et de la composante bleue, d'autre part et la valeur non filtrée desdites composantes respectivement, est supérieure à 2, ledit bit additionnel est positionné à 1.

10. Procédé selon la revendication 9, **caractérisé en ce que** si la différence entre la valeur filtrée de la composante rouge, d'une part, et de la composante bleue, d'autre part et la valeur non filtrée desdites composantes, respectivement, est inférieure ou égale à - 2, le bit additionnel obtenu en soustrayant le bit 1 à l'endroit du bit additionnel.

11. Procédé selon la revendication 10, **caractérisé en ce que** si ladite différence est inférieure ou égale à 2 ou supérieure ou égale à - 2, ledit bit additionnel est positionné à 0.

12. Procédé selon l'une quelconque des revendications 1 à 11, **caractérisé en ce que** l'on compare la valeur d'un pixel en cours de traitement et de pixels adjacents avec une valeur de seuil et l'on affecte au pixel en cours de traitement une valeur maximale si ce pixel et les pixels adjacents ont tous une valeur supérieure à la valeur de seuil.

13. Dispositif de traitement d'image pour la conversion d'un signal d'un premier nombre de bits par pixel en un signal d'un deuxième nombre de bits par pixel supérieur au premier nombre de bits par pixel, le signal à convertir et le signal converti comprenant un ensemble de pixels associés chacun à des valeurs de composantes de rouge, de vert et de bleu constituant une trame dudit premier nombre de bits et une trame dudit deuxième nombre de bits, respectivement, **caractérisé en ce qu'il** comporte des moyens (30) pour élaborer des valeurs de bits additionnels de codage des composantes de chaque pixel à partir des valeurs de composantes de pixels adjacents et des moyens (30) pour élaborer le signal converti à partir des bits de codage des valeurs des composantes de rouge, de vert et de bleu du signal à convertir et desdits bits additionnels.

14. Ecran plat, comprenant un ensemble de pixels commandés par un ou plusieurs circuits de commande périphérique convertissant un signal d'un premier nombre de bits par pixel en un signal d'un deuxième nombre de bits par pixel supérieur au premier nombre de bits par pixel, le signal à convertir et le signal converti comprenant un ensemble de pixels associés chacun à des valeurs de composantes de rouge, de vert et de bleu constituant une trame dudit premier nombre de bits et une trame dudit deuxième nombre de bits, respectivement, **caractérisé en ce qu'il** comporte des moyens (30) pour élaborer des valeurs de bits additionnels de codage des composantes de chaque pixel à partir des valeurs de composantes de pixels adjacents et des moyens (30) pour élaborer le signal converti à partir des bits de codage des valeurs des composantes de rouge, de vert et de bleu du signal à convertir et desdits bits additionnels.

15. Equipement électronique portable, du type téléphone mobile, **caractérisé en ce qu'il** comporte un écran plat selon la revendication 14.

16. Equipement selon la revendication 15, **caractérisé en ce qu'il** constitue un téléphone mobile.

5

10

15

20

25

30

35

40

45

50

55

FIG.1

@	31...27	26...21	20...16	15...11	10...5	4...0
0	R _{n+1}	V _{n+1}	B _{n+1}	R _n	V _n	B _n
4	R _{n+3}	V _{n+3}	B _{n+3}	R _{n+2}	V _{n+2}	B _{n+2}

FIG.2

@	31...24	23...16	15...8	7...0
0	B _{n+1}	R _n	V _n	B _n
4	V _{n+2}	B _{n+2}	R _{n+1}	V _{n+1}
8	R _{n+3}	V _{n+3}	B _{n+3}	R _{n+2}

FIG.3

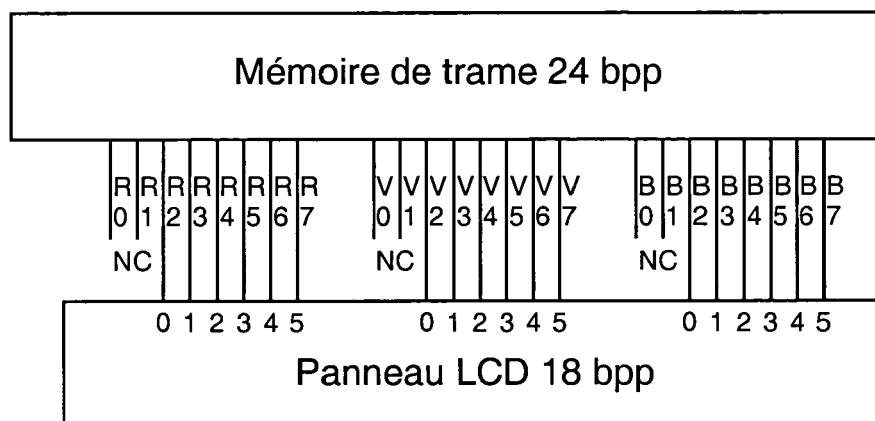


FIG.4

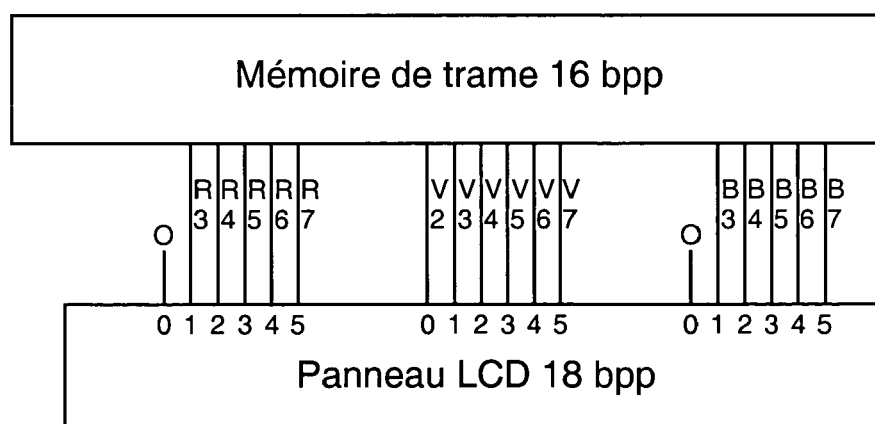


FIG.5

$R_{i-2,j-2}, B_{i-2,j-2}$ $V_{i-2,j-2}$	$R_{i-2,j-1}, B_{i-2,j-1}$ $V_{i-2,j-1}$	$R_{i-2,j}, B_{i-2,j}$ $V_{i-2,j}$	$R_{i-2,j+1}, B_{i-2,j+1}$ $V_{i-2,j+1}$	$R_{i-2,j+2}, B_{i-2,j+2}$ $V_{i-2,j+2}$
$R_{i-1,j-2}, B_{i-1,j-2}$ $V_{i-1,j-2}$	$R_{i-1,j-1}, B_{i-1,j-1}$ $V_{i-1,j-1}$	$R_{i-1,j}, B_{i-1,j}$ $V_{i-1,j}$	$R_{i-1,j+1}, B_{i-1,j+1}$ $V_{i-1,j+1}$	$R_{i-1,j+2}, B_{i-1,j+2}$ $V_{i-1,j+2}$
$R_{i,j-2}, B_{i,j-2}$ $V_{i,j-2}$	$R_{i,j-1}, B_{i,j-1}$ $V_{i,j-1}$	$R_{i,j}, B_{i,j}$ $V_{i,j}$	$R_{i,j+1}, B_{i,j+1}$ $V_{i,j+1}$	$R_{i,j+2}, B_{i,j+2}$ $V_{i,j+2}$
$R_{i+1,j-2}, B_{i+1,j-2}$ $V_{i+1,j-2}$	$R_{i+1,j-1}, B_{i+1,j-1}$ $V_{i+1,j-1}$	$R_{i+1,j}, B_{i+1,j}$ $V_{i+1,j}$	$R_{i+1,j+1}, B_{i+1,j+1}$ $V_{i+1,j+1}$	$R_{i+1,j+2}, B_{i+1,j+2}$ $V_{i+1,j+2}$
$R_{i+2,j-2}, B_{i+2,j-2}$ $V_{i+2,j-2}$	$R_{i+2,j-1}, B_{i+2,j-1}$ $V_{i+2,j-1}$	$R_{i+2,j}, B_{i+2,j}$ $V_{i+2,j}$	$R_{i+2,j+1}, B_{i+2,j+1}$ $V_{i+2,j+1}$	$R_{i+2,j+2}, B_{i+2,j+2}$ $V_{i+2,j+2}$

P

F

FIG.6

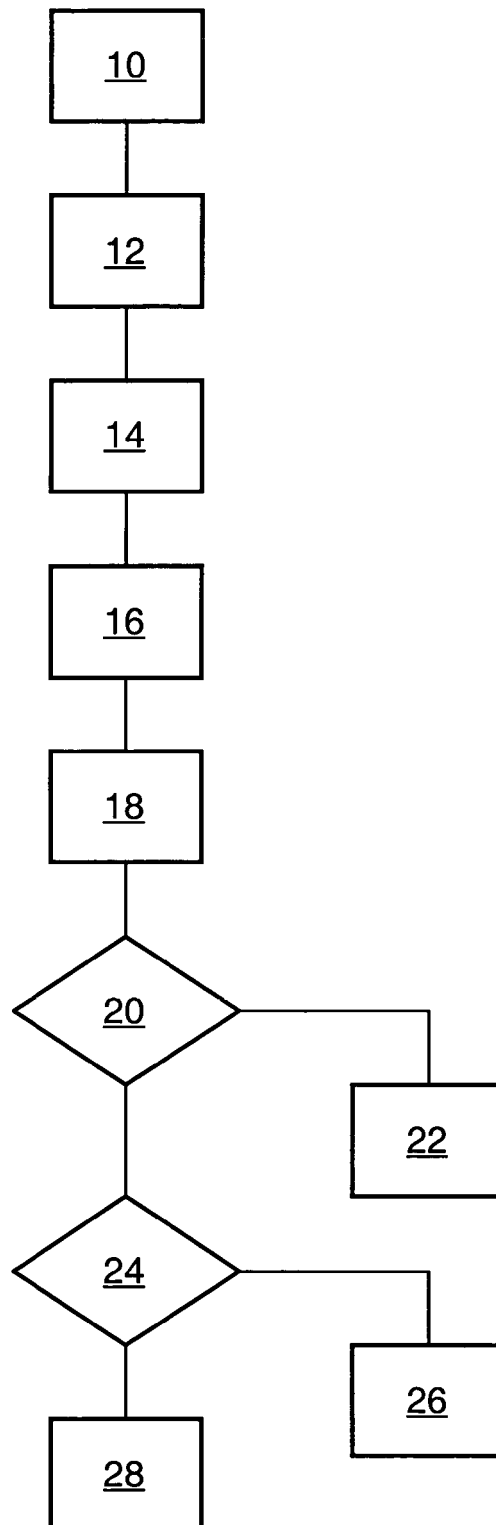


FIG.7

$R_{i-1,j-1} - V_{i-1,j-1}$	$R_{i-1,j} - V_{i-1,j}$	$R_{i-1,j+1} - V_{i-1,j+1}$
$R_{i,j-1} - V_{i,j-1}$	$R_{i,j} - V_{i,j}$	$R_{i,j+1} - V_{i,j+1}$
$R_{i+1,j-1} - V_{i+1,j-1}$	$R_{i+1,j} - V_{i+1,j}$	$R_{i+1,j+1} - V_{i+1,j+1}$

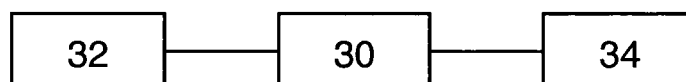
FIG.8

1	2	1
2	4	2
1	2	1

$C_{i,j}$

FIG.9

	Mémoire de trame de 24 bpp	Mémoire de trame 16 bpp rouge lsb = bleu lsb = 0		Conversion 16 bpp 18 bpp
		Troncature	arrondi	
PSNR rouge	42,69 dB	35,70 dB	40,72 dB	42,00 dB
PSNR vert	42,69 dB	42,69 dB	46,37 dB	46,37 dB
PSNR bleu	42,69 dB	35,70 dB	40,72 dB	42,00 dB
PSNR luma	42,69 dB	38,45 dB	43,14 dB	44,00 dB
Taille T de mémoire de trame	T x 1,5	T	T	T

FIG.10



DOCUMENTS CONSIDERES COMME PERTINENTS			
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes	Revendication concernée	CLASSEMENT DE LA DEMANDE (IPC)
X	US 2003/231195 A1 (UENO SATOSHI ET AL) 18 décembre 2003 (2003-12-18) * abrégé * * alinéas [0008] - [0010], [0079] - [0105]; figures 1-3 *	1-5, 13-16	INV. G09G5/04 G09G3/20
A	US 2002/109702 A1 (KOBAYASHI KAZUO) 15 août 2002 (2002-08-15) * abrégé * * alinéas [0011] - [0018], [0031] - [0038]; figures 3,5 *	1,13-16	
A	US 2003/169244 A1 (KUROKAWA YASUHIRO ET AL) 11 septembre 2003 (2003-09-11) * abrégé * * alinéa [0054] - alinéa [0058]; figure 8 *	1,13-16	
A	EP 1 363 267 A (MITSUBISHI DENKI KABUSHIKI KAISHA) 19 novembre 2003 (2003-11-19) * abrégé * * alinéa [0049] - alinéa [0060]; figures 5-8 *	7,8	DOMAINES TECHNIQUES RECHERCHES (IPC) G09G
Le présent rapport a été établi pour toutes les revendications			
Lieu de la recherche Munich		Date d'achèvement de la recherche 24 mai 2006	Examineur Wolff, L
CATEGORIE DES DOCUMENTS CITES X : particulièrement pertinent à lui seul Y : particulièrement pertinent en combinaison avec un autre document de la même catégorie A : arrière-plan technologique O : divulgation non-écrite P : document intercalaire		T : théorie ou principe à la base de l'invention E : document de brevet antérieur, mais publié à la date de dépôt ou après cette date D : cité dans la demande L : cité pour d'autres raisons & : membre de la même famille, document correspondant	

EPO FORM 1503 03.82 (P04C02)

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

EP 06 29 0242

La présente annexe indique les membres de la famille de brevets relatifs aux documents brevets cités dans le rapport de recherche européenne visé ci-dessus.

Lesdits membres sont contenus au fichier informatique de l'Office européen des brevets à la date du

Les renseignements fournis sont donnés à titre indicatif et n'engagent pas la responsabilité de l'Office européen des brevets.

24-05-2006

Document brevet cité au rapport de recherche		Date de publication	Membre(s) de la famille de brevet(s)	Date de publication
US 2003231195	A1	18-12-2003	CN 1468006 A	14-01-2004
			JP 3710131 B2	26-10-2005
			JP 2004054210 A	19-02-2004
			TW 223565 B	01-11-2004

US 2002109702	A1	15-08-2002	JP 2002140030 A	17-05-2002

US 2003169244	A1	11-09-2003	JP 2003330433 A	19-11-2003

EP 1363267	A	19-11-2003	CN 1461140 A	10-12-2003
			JP 2003330447 A	19-11-2003
			TW 225361 B	11-12-2004
			US 2003215132 A1	20-11-2003

EPO FORM P0460

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