(19) RÉPUBLIQUE FRANÇAISE

INSTITUT NATIONAL
DE LA PROPRIÉTÉ INDUSTRIELLE

PARIS

(11) **N° de publication :** **2 803 156**
(à n'utiliser que pour les commandes de reproduction)

(21) **N° d'enregistrement national :** **99 16250**

(51) Int Cl[7] : **H 04 N 1/00,** H 04 N 1/64, 1/46

(12) DEMANDE DE BREVET D'INVENTION A1

(22) **Date de dépôt :** 22.12.99.

(30) **Priorité :**

(43) **Date de mise à la disposition du public de la demande :** 29.06.01 Bulletin 01/26.

(56) **Liste des documents cités dans le rapport de recherche préliminaire :** *Se reporter à la fin du présent fascicule*

(60) **Références à d'autres documents nationaux apparentés :**

(71) **Demandeur(s) :** *SAGEM SA Société anonyme* — FR.

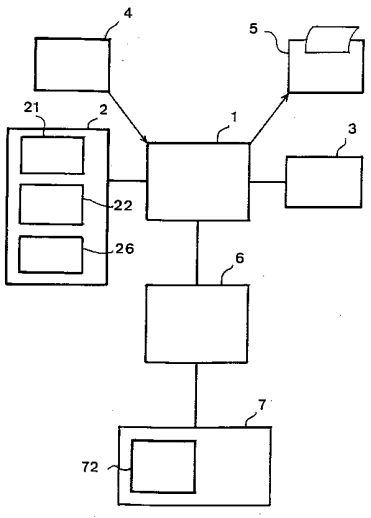
(72) **Inventeur(s) :** CREVOISIER DE STANISLAS.

(73) **Titulaire(s) :**

(74) **Mandataire(s) :** CABINET BLOCH.

(54) TELECOPIEUR COULEUR COMPORTANT UN MICROPROCESSEUR SECONDAIRE DE TRAITEMENT D'IMAGES.

(57) Le télécopieur couleur comprend des circuits de traitement comportant un microprocesseur central (1) avec au moins une mémoire de programmes de traitement (2) et une mémoire de travail (3), télécopieur et il comprend en outre un microprocesseur secondaire (6) de traitement de données couleur associé au microprocesseur central (1) et à une mémoire de travail dédiée (7) agencée pour recevoir des tables de changement d'espace couleurs, le microprocesseur secondaire couleur (6) étant commandé par des programmes stockés dans la mémoire de programmes (2) du microprocesseur central (1).



FR 2 803 156 - A1

Pour télécopier une image en couleur, il faut d'abord l'analyser, ou la scanner, et, avant de la transmettre en ligne, il faut la coder, la traiter. Inversement, pour recevoir une télécopie en couleur, il faut d'abord procéder à un traitement des données de l'image reçues, avant d'imprimer l'image. Un télécopieur classique comporte aussi bien un scanner qu'une imprimante. Par contre, pour le traitement d'image, il faut aujourd'hui faire usage d'un ordinateur, par exemple un terminal PC, d'ailleurs équipé d'un modem, qu'on associe donc à un télécopieur pour proposer de la télécopie couleur.

La présente invention vise à s'affranchir d'un ordinateur dans le traitement des télécopies couleur et propose, à cet effet, un télécopieur couleur comprenant des moyens de traitement comportant un microprocesseur central avec au moins une mémoire de programmes de traitement et une mémoire de travail, télécopieur caractérisé par le fait qu'il comprend en outre un microprocesseur secondaire de traitement de données couleur associé au microprocesseur central et à une mémoire de travail dédiée agencée pour recevoir des tables de changement d'espace couleurs, le microprocesseur secondaire couleur étant agencé pour être commandé par des programmes stockés dans la mémoire de programmes du microprocesseur central.

Dans un équipement de traitement de données couleur, de nombreux changements d'espace de représentation des couleurs sont souvent nécessaires pour adapter au mieux l'espace au traitement.

Ainsi, et à titre d'exemple, un document peut être scanné en RVB (rouge, vert, bleu), affiché sous PC en SRVB (s pour standard), comprimé en mode CIELab (espace luminance, axe a, axe b, du Comité International de l'Eclairage) transmis en mode YCrCb (mode Lab version TB) et imprimé en mode CMJN (Cyan, Magenta, Jaune, Noir).

Les espaces de représentation associés à un scanner et à une imprimante sont des espaces dits dépendants, car ils dépendent des caractéristiques structurelles de ces éléments. Les autres espaces sont des espaces indépendants qui ne dépendent d'aucun élément matériel. Dans un

équipement de traitement de données couleurs, on utilise donc des tables de passage d'espaces dépendants vers des espaces indépendants et inversement.

Stocker toutes les tables de changement d'espace de représentation des couleurs dans la mémoire de programmes du microprocesseur central nécessiterait une mémoire de taille prohibitive.

Aussi, dans la forme de réalisation préférée du télécopieur couleur de l'invention, le microprocesseur central est agencé pour, à l'initialisation du télécopieur, calculer toutes les tables de changement d'espace couleur et stocker, dans la mémoire de programmes de traitement du microprocesseur central, la table de passage de l'espace dépendant de représentation des couleurs du scanner vers un espace indépendant de représentation et la table de passage d'un autre espace indépendant de représentation vers l'espace dépendant de représentation des couleurs de l'imprimante et, dans la mémoire de travail dédiée du microprocesseur secondaire couleur, toutes les autres tables de changement d'espace couleur.

L'invention sera mieux comprise à l'aide de la description suivante d'une forme de réalisation préférée d'un télécopieur couleur selon l'invention, en référence à la figure unique annexée, qui en est un schéma par blocs et en référence au programme CEC.cpp de changement d'espace couleur en annexe.

Le télécopieur représenté comporte un microprocesseur central 1 relié à une mémoire morte ROM 2 de programmes de traitement, à une mémoire vive RAM de travail, à un scanner 4 et à une imprimante 5. Les circuits classiques de relations homme-machine et de transmission des télécopies n'ont pas été représentés.

Au microprocesseur 1 est en outre relié un microprocesseur secondaire 6 associé, de traitement d'images et, notamment, de données couleur, lui-même relié à une mémoire vive de travail RAM 7 dédiée, prévue pour recevoir des tables 72 de changement d'espace couleurs. Le

microprocesseur 1 est concerné par toutes les fonctions autres que le traitement d'images. Le microprocesseur secondaire couleur 6 est commandé par des programmes, ou codes de commandes, stockés dans une zone 26 de la mémoire de programmes 2 du microprocesseur central 1. La représentation du dessin n'est que schématique et, en pratique, un bus relie les divers circuits représentés et offre ainsi une possibilité de liaison directe entre ceux-ci.

Comme évoqué au début, le télécopieur doit pouvoir effectuer un grand nombre de changements d'espace de représentation des couleurs, chacun adapté à une opération déterminée : analyse par le scanner 4, dans l'espace dépendant qui lui est propre, compression, transmission, impression par l'imprimante 5 dans un autre espace dépendant adapté à celle-ci.

Les espaces dépendants liés à un dispositif, scanner 4 (RVB) et imprimante 5 (CMJN), sont spécifiés, dans une zone 21 de la mémoire morte ROM 2, par une matrice ou table de transformation, de changement d'espace couleur, permettant de faire passer des données d'image couleur de l'espace dépendant considéré à un espace indépendant (CIELab, YCrCb et autres), ou inversement. Deux espaces indépendants quelconques sont mutuellement liés par une table de transformation qui, indépendante de tout élément matériel, a été conçue pour ne pas présenter de distorsions et a donc pu être définie par un algorithme.

De ce fait, seules sont stockées en mémoire morte 2, dans la zone 21 de changement d'espace couleur, les tables de transformation mettant en jeu au moins un espace dépendant (RVB, CMJN), c'est-à-dire la table de passage de l'espace dépendant de représentation des couleurs du scanner 4 vers un espace indépendant de représentation d'image et la table de passage d'un autre espace indépendant de représentation d'image vers l'espace dépendant de représentation des couleurs de l'imprimante 5. A titre d'exemple, le programme CEC.cpp en annexe fait passer les données de l'espace indépendant Lab à l'espace CMJN. Toutes les autres tables, de transformation entre espaces indépendants (CIELab, YCrCb), sont représentées, dans une zone 22 de la mémoire 2, par uniquement

l'algorithme permettant de les reconstituer par calcul. En pratique ici, ce calcul de toutes les tables de changement d'espace couleur indépendant intervient systématiquement, à l'initialisation du télécopieur, et les tables correspondantes sont stockées dans la zone 72 de la mémoire RAM 7. Le microprocesseur 6 dispose ainsi de ces tables en temps réel.

Il est aussi prévu de calculer et stocker de même une table redondante représentant globalement une matrice de passage direct de l'espace dépendant RVB du scanner 4 à l'espace dépendant CMJN de l'imprimante 5, afin de limiter la tâche de traitement en cas de copie locale.

Un espace de référence de couleurs comporte ici 256 valeurs par composante de référence. L'espace RVB par exemple peut ainsi être représenté par un cube d'arête de longueur 256, à trois axes, R, V, B.

Le passage de données de couleur d'un premier espace à un second espace s'effectue en trois étapes. Dans une première étape, on divise chaque axe du premier espace, ici R, V, B, en $2^N + 1$ segments, par exemple en 33 segments si N=5, ce qui définit 33^3 volumes élémentaires cubiques d'arête égale à 8 unités. Dans une deuxième étape, on détermine par des tables ou algorithmes de transformation, telles que rotation et translation ou autres, les positions des sommets des cubes élémentaires du premier espace dans le second espace visé. On obtient ainsi des volumes élémentaires correspondants dans le second espace, dont la forme peut différer de celle d'un cube. La deuxième étape permet ainsi de placer tout point du premier espace, entraîné avec le cube le contenant, dans une position approximative correspondante du second espace. Dans une troisième étape, la position exacte du point du premier espace transféré dans le second espace est déterminée, à partir de tables en mémoire morte 2, par interpolation tétraédrique d'après la position du point dans le cube de départ le contenant. Pour ce faire, la position relative du point (3 valeurs de 0 à 8 selon les 3 axes respectifs R, V, B) étant repérée par rapport aux sommets du cube, il est déterminé une position relative homologue par rapport aux sommets de l'espace élémentaire constituant la transformée du cube considéré. On notera que, d'une façon générale, les espaces considérés sont des espaces vectoriels dont le nombre de vecteurs

propres peut être quelconque et que les volumes élémentaires tridimensionnels de l'exemple ci-dessus pourraient, dans un autre exemple, avoir un autre nombre de dimensions. Un volume élémentaire peut donc être défini comme étant une portion de l'espace vectoriel considéré, dans laquelle toutes les variables, spécifiant des positions par rapport aux vecteurs propres, varient mais ne présentent qu'une variation élémentaire.

```
////////////////////////////////////////////////////////////////
//                                        CEC.cpp
            //
//
            //
// Ce programme effectue un changement d'espace                    //
// colorimétrique Lab -> CMJN à l'aide de tables                   //
// d'interpolation codées sur 8 bits.                                    //
//
            //
// Par Stanislas de Crevoisier, octobre 1998 (Stan3.cpp)    //
// Modifié le 04/03/99 pour le module "imprimante", SdC            //
// Modifié le 15/10/99 pour le projet "calibration", SdC    //
////////////////////////////////////////////////////////////////

#include "stdafx.h"
#include <math.h>
#include "calibration.h"
#include "imprimante.h"
#include "conversion.h"
#include "TablesCEC.h"

//Variables globales
typedef struct taglpcci
{
      int offset[10][4];
      unsigned short lpCoeff[32768][5];
      unsigned short lpProfil[35937][4];
} CCECONVERTINFO, *LPCCECONVERTINFO;

// Eléments relatifs à la grille fine (LUT complète)
unsigned char pas;
unsigned char taille_cube;
unsigned char taille_imp,taille_scan;
unsigned char lsb, lsb2, msb,mask_lsb;
unsigned short n_coins_1D, n_coins_2D;
//CString chemin_coeffs="C:\\Program
Files\\DevStudio\\MyProjects\\Calibration\\Aide à la calibration\\";
CString chemin_coeffs;

//Détermination des variables globales
void calculer_variables(void)
{
      switch(taille_imp)
      {
      case 9 :
            lsb = 5;
            break;
      case 17 :
            lsb = 4;
            break;
      case 33 :
            lsb = 3;
            break;
      }

      lsb2 = (lsb << 1);
      msb = 8 - lsb;
      n_coins_1D = taille_imp;
      n_coins_2D = taille_imp * taille_imp;
      mask_lsb = (1<<lsb) - 1;
```

```
	taille_cube = mask_lsb + 1;
}

//chargement des profils Lab 8 bits
LPCCECONVERTINFO charger_profils(LPCCECONVERTINFO lpcci, CString nom_fichier)
{
	int a,b,taille;
	FILE *fichier;

	fichier=fopen(nom_fichier, "rb");
	if(fichier==NULL)
	{
		CString strMsg;
		strMsg="Erreur d'ouverture du profil imprimante";
		MessageBox(NULL,strMsg, NULL, MB_ICONINFORMATION | MB_OK);
		return 0;
	}

	//On récupère la taille des tables
	fseek(fichier, 0L, SEEK_END);
	taille = ftell(fichier);
	fseek(fichier, 0L, SEEK_SET);

	switch(taille)
	{
	case 2916 :
		taille_imp = 9;
		break;
	case 19652 :
		taille_imp = 17;
		break;
	case 143748 :
		taille_imp = 33;
		break;
	default :
		CString strMsg;
		strMsg="Erreur de taille des tables imprimante";
		MessageBox(NULL,strMsg, NULL, MB_ICONINFORMATION | MB_OK);
		return 0;
	}

	calculer_variables();

	for(a=0;a<(taille_imp*taille_imp*taille_imp);a++)
	{
		for(b=0;b<4;b++)
		{
			lpcci->lpProfil[a][b]=fgetc(fichier);
		}
	}
	fclose(fichier);

	return lpcci;
}

//chargement des coefficients 16 bits
LPCCECONVERTINFO charger_coeffs(LPCCECONVERTINFO lpcci)
{
	int a,b;
	CString nom_fichier;
	FILE *fichier;
```

```
	//fichier=fopen("Coefs3.don", "r");
	//fichier=fopen("Coeff16b.don", "r");
	switch(taille_imp)
	{
	case 9 :
		nom_fichier = chemin_coeffs + "Coeffs9x9x9.don";
		break;
	case 17 :
		nom_fichier = chemin_coeffs + "Coeffs17x17x17.don";
		break;
	case 33 :
		nom_fichier = chemin_coeffs + "Coeffs33x33x33.don";
		break;
	}

	fichier = fopen(nom_fichier,"rb");

	if(fichier==NULL)
	{
		CString strMsg;
		strMsg="Erreur d'ouverture du fichier de coefficients";
		MessageBox(NULL,strMsg, NULL, MB_ICONINFORMATION | MB_OK);
		return 0;
	}

	for(a=0;a<(1<<(lsb*3));a++)
	{
		for(b=0;b<5;b++)
		{
			lpcci->lpCoeff[a][b]=fgetc(fichier);
			lpcci->lpCoeff[a][b]+=(fgetc(fichier)<<8);
		}
	}
	fclose(fichier);

	return lpcci;
}

//procédure de détermination des offsets

LPCCECONVERTINFO charger_offsets(LPCCECONVERTINFO lpcci)
{
	int q  = n_coins_1D;
	int offset[8];
	int type[10];

	// initialisations pour trouver les offsets
	// Il est plus rapide de calculer une table pour toutes les combinaisons
	// Lors du traitement de chaque pixel on fait 4 indirections
	// à la place de 5 indirections et 8 tests de masquage !
	offset[0] = 0;		// coordonnées de A dans le cube
	offset[1] = 1;		// coordonnées de B dans le cube
	offset[2] = q+1;	// coordonnées de C dans le cube
	offset[3] = q;		// coordonnées de D dans le cube
	offset[4] = q*q;	// coordonnées de E dans le cube
	offset[5] = q*q+1;	// coordonnées de F dans le cube
	offset[6] = q*q+q+1;	// coordonnées de G dans le cube
	offset[7] = q*q+q;	// coordonnées de H dans le cube

	type[0] = 0x1b;
```

```
	type[1] = 0x72;
	type[2] = 0x4e;
	type[3] = 0xd8;
	type[4] = 0x5a;
	type[5] = 0xb1;
	type[6] = 0x27;
	type[7] = 0xe4;
	type[8] = 0x8d;
	type[9] = 0xa5;

	for (int nType = 0; nType < 10; nType++)
	{
		char tetra = type[nType];
		int m = 1;
		int nOffset = 0;
		for (int sommet = 0; sommet < 8; sommet ++, m <<= 1)
		{
			if (tetra & m)
			{
				lpcci->offset[nType][nOffset] = offset[sommet];
				nOffset ++;
			}
		}
	}
	return lpcci;
}

struct CMJNQUADRUPLE calcul(LPCCECONVERTINFO lpcci,struct LABTRIPLE Lab)
{
	int L,a,b;
	struct CMJNQUADRUPLE CMJN;
	int i,j,tmp;

	long x,aux[4];
	int sommets,offsetCoef,type,coef;
	unsigned short lpCoeff[4];
	unsigned short lpProfil[4];   //+pratiques que les pointeurs...

	//Boucle écrite par CCE

	for(i=0;i<1;i++) //Fausse boucle pour traitements ultérieurs.
	{
		L=(int)(Lab.L);
		a=(int)(Lab.a);
		b=(int)(Lab.b);
		aux[0]=0;
		aux[1]=0;
		aux[2]=0;
		aux[3]=0;
		offsetCoef = (L & mask_lsb) | ((a & mask_lsb) << lsb) | ((b &
mask_lsb) << lsb2);
		x=(long) n_coins_2D*(b>>lsb) + n_coins_1D*(a>>lsb) + (L>>lsb);
		for(j=0;j<4;j++)
			lpCoeff[j] = lpcci->lpCoeff[offsetCoef][j];
		type=lpcci->lpCoeff[offsetCoef][4];
		for(sommets=0;sommets<4;sommets++)
		{
			tmp=(lpcci->offset[type][sommets]+x);
			for(j=0;j<4;j++)
			{
				lpProfil[j]=lpcci->lpProfil[tmp][j];
```

```
                }
                coef=lpCoeff[sommets];
                aux[0] += (long)(lpProfil[0]*coef);
                aux[1] += (long)(lpProfil[1]*coef);
                aux[2] += (long)(lpProfil[2]*coef);
                aux[3] += (long)(lpProfil[3]*coef);
            }
            CMJN.c = (unsigned char)(aux[0]>>15);
            CMJN.m = (unsigned char)(aux[1]>>15);
            CMJN.j = (unsigned char)(aux[2]>>15);
            CMJN.n = (unsigned char)(aux[3]>>15);
      }

      return CMJN;
}

//Permet d'appeler les différentes fonctions définies plus haut...
//Réalise la conversion de tables :
//RVB->Lab et Lab->CMJN en RVB->CMJN
CString cec(CString nom_profil_scanner, CString nom_profil_imp)
{
      FILE *fic1, *fic2, *fic_sRGB, *fic_YCC;
      LPCCECONVERTINFO lpcci = new(CCECONVERTINFO);
      int r,v,b;
      unsigned int valeur[3];
      struct RGBTRIPLE2 RGB, XYZ;
      struct LABTRIPLE Lab;
      struct CMJNQUADRUPLE CMJN;
      CString nom_profil_copie_locale;
      CString nom_profil_cmjn_srgb, nom_profil_cmjn_ycc;

      chemin_coeffs.LoadString(IDS_BASE_DIRECTORY);

      //détermination de la structure lpcci
      //printf("Initialisation du changement d'espace colorimetrique...");
      lpcci=charger_profils(lpcci,nom_profil_imp);
      if(lpcci == NULL)
            return "erreur";
      lpcci=charger_coeffs(lpcci);
      if(lpcci == NULL)
            return "erreur";
      lpcci=charger_offsets(lpcci);
      if(lpcci == NULL)
            return "erreur";

      //chargement de la LUT 16 bits RVB->Lab
      //printf("OK\nChargement du profil scanner...\n");
      if((fic1=fopen(nom_profil_scanner,"rb"))==NULL)
      {
            CString strMsg;
            strMsg="Erreur d'ouverture du profil scanner";
            MessageBox(NULL,strMsg, NULL, MB_ICONINFORMATION | MB_OK);
            return "erreur";
      }

      //On récupère la taille des tables
      fseek(fic1, 0L, SEEK_END);
      r = ftell(fic1);
      fseek(fic1, 0L, SEEK_SET);

      switch(r)
```

```
{
case 2187 :
      taille_scan = 9;
      break;
case 14739 :
      taille_scan = 17;
      break;
case 107811 :
      taille_scan = 33;
      break;
default :
      CString strMsg;
      strMsg="Erreur de taille des tables du profil scanner";
      MessageBox(NULL,strMsg, NULL, MB_ICONINFORMATION | MB_OK);
      return "erreur";
}

nom_profil_copie_locale = nom_profil_scanner.SpanExcluding(".");
nom_profil_cmjn_srgb = nom_profil_scanner.SpanExcluding("_");
nom_profil_cmjn_ycc = nom_profil_cmjn_srgb + "_ycc_cmjn.raw";
nom_profil_cmjn_srgb = nom_profil_cmjn_srgb + "_srgb_cmjn.raw";
nom_profil_copie_locale += "_cmjn.raw";

if((fic2=fopen(nom_profil_copie_locale,"wb"))==NULL)
{
      CString strMsg;
      strMsg="Erreur d'ouverture du profil copie locale";
      MessageBox(NULL,strMsg, NULL, MB_ICONINFORMATION | MB_OK);
      return "erreur";
}

if((fic_sRGB=fopen(nom_profil_cmjn_srgb,"wb"))==NULL)
{
      CString strMsg;
      strMsg="Erreur d'ouverture du profil sRGB -> CMJN";
      MessageBox(NULL,strMsg, NULL, MB_ICONINFORMATION | MB_OK);
      return "erreur";
}

if((fic_YCC=fopen(nom_profil_cmjn_ycc,"wb"))==NULL)
{
      CString strMsg;
      strMsg="Erreur d'ouverture du profil YCrCb -> CMJN";
      MessageBox(NULL,strMsg, NULL, MB_ICONINFORMATION | MB_OK);
      return "erreur";
}

//printf("OK\nCreation de la table RVB->CMJN...");
for(r=0;r<(taille_scan*taille_scan*taille_scan);r++)
{
      for(b=0;b<3;b++)
      {
            valeur[b]=fgetc(fic1);
            ////valeur[b]+=(fgetc(fic1)<<8);
      }
      RGB.red = valeur[0];
      RGB.green = valeur[1];
      RGB.blue = valeur[2];
      XYZ = sRGB2XYZ(RGB);
      //Lab.L=(float)valeur[0];
      //Lab.a=(float)valeur[1];
```

```
        //Lab.b=(float)valeur[2];
        Lab = XYZ2Lab(XYZ.red,XYZ.green,XYZ.blue);
        Lab.L = (float)(Lab.L*2.55);
        if(Lab.L > 255)
              Lab.L = 255;
        if(Lab.L<0)
              Lab.L = 0;
        Lab.a = (float)((Lab.a+85)*255.0/170.0);
        Lab.b = (float)((Lab.a+75)*255.0/200.0);
        if(Lab.a > 255)
              Lab.a = 255;
        if(Lab.a<0)
              Lab.a = 0;
        if(Lab.b > 255)
              Lab.b = 255;
        if(Lab.b<0)
              Lab.b = 0;
        CMJN=calcul(lpcci,Lab);
        fputc(CMJN.c,fic2);
        fputc(CMJN.m,fic2);
        fputc(CMJN.j,fic2);
        fputc(CMJN.n,fic2);
}

//Création des tables sRGB -> CMJN et YCC -> CMJN
//Elles sont 33x33x33 par défaut
for(b=0;b<257;b+=8)
{
      if(b == 256)
            b = 255;
      for(v=0;v<257;v+=8)
      {
            if(v == 256)
                  v = 255;
            for(r=0;r<257;r+=8)
            {
                  if(r == 256)
                        r = 255;

                  RGB.red = r;
                  RGB.green = v;
                  RGB.blue = b;

                  //sRGB -> CMJN
                  XYZ = sRGB2XYZ(RGB);
                  Lab = XYZ2Lab(XYZ.red, XYZ.green, XYZ.blue);
                  CMJN=calcul(lpcci,Lab);
                  fputc(CMJN.c,fic_sRGB);
                  fputc(CMJN.m,fic_sRGB);
                  fputc(CMJN.j,fic_sRGB);
                  fputc(CMJN.n,fic_sRGB);

                  //YCC -> CMJN
                  RGB = YCrCb2sRGB(RGB);
                  XYZ = sRGB2XYZ(RGB);
                  Lab = XYZ2Lab(XYZ.red, XYZ.green, XYZ.blue);
                  CMJN=calcul(lpcci,Lab);
                  fputc(CMJN.c,fic_YCC);
                  fputc(CMJN.m,fic_YCC);
                  fputc(CMJN.j,fic_YCC);
                  fputc(CMJN.n,fic_YCC);
```

```
                }
            }
        }
        fclose(fic1);
        fclose(fic2);
        fclose(fic_sRGB);
        fclose(fic_YCC);

        return nom_profil_copie_locale;
}

//RVB->Lab et Lab->CMJN en RVB->CMJN
int format_tables_imp(CString nom_profil_imp)
{
        FILE *fichier;
        LPCCECONVERTINFO lpcci = new(CCECONVERTINFO);
        int r,v,b;
        struct LABTRIPLE Lab;
        struct CMJNQUADRUPLE CMJN;
        //RGBTRIPLE2 RGB; //////

        chemin_coeffs.LoadString(IDS_BASE_DIRECTORY);

        //détermination de la structure lpcci
        //printf("Initialisation du changement d'espace colorimetrique...");
        lpcci=charger_profils(lpcci,nom_profil_imp);
        if(lpcci == NULL)
                return 0;
        lpcci=charger_coeffs(lpcci);
        if(lpcci == NULL)
                return 0;
        lpcci=charger_offsets(lpcci);
        if(lpcci == NULL)
                return 0;

        if((fichier=fopen(nom_profil_imp,"wb"))==NULL)
        {
                CString strMsg;
                strMsg="Erreur d'ouverture du profil imprimante";
                MessageBox(NULL,strMsg, NULL, MB_ICONINFORMATION | MB_OK);
                return 0;
        }

        //Création des tables Lab->CMJN au bon format
        for(b=0;b<257;b+=taille_cube)
        {
                if(b == 256)
                        b = 255;
                for(v=0;v<257;v+=taille_cube)
                {
                        if(v == 256)
                                v = 255;
                        for(r=0;r<257;r+=taille_cube)
                        {
                                if(r == 256)
                                        r = 255;

                                //RGB.red = r;/////
                                //RGB.green = v;/////
                                //RGB.blue = b;/////
                                //RGB = sRGB2XYZ(RGB);/////
```

```
                        //Lab = XYZ2Lab(RGB.red, RGB.green, RGB.blue);/////
                        Lab.L = (float)r;
                        Lab.a = (float)v;
                        Lab.b = (float)b;
                        Lab = conversionOasis_ICC(Lab);
                        CMJN=calcul(lpcci,Lab);
                        fputc(CMJN.c,fichier);
                        fputc(CMJN.m,fichier);
                        fputc(CMJN.j,fichier);
                        fputc(CMJN.n,fichier);
                    }
                }
            }
            fclose(fichier);

            return 1;
        }
```

REVENDICATIONS

1.- Télécopieur couleur comprenant des moyens de traitement comportant un microprocesseur central (1) avec au moins une mémoire de programmes de traitement (2) et une mémoire de travail (3), télécopieur caractérisé par le fait qu'il comprend en outre un microprocesseur secondaire (6) de traitement de données couleur associé au microprocesseur central (1) et à une mémoire de travail dédiée (7) agencée pour recevoir des tables de changement d'espace couleurs, le microprocesseur secondaire couleur (6) étant agencé pour être commandé par des programmes stockés dans la mémoire de programmes (2) du microprocesseur central (1).

2.- Télécopieur selon la revendication 1, dans lequel le microprocesseur central (1) est agencé pour, à l'initialisation du télécopieur, calculer toutes les tables de changement d'espace couleur et stocker, dans la mémoire (2) de programmes de traitement du microprocesseur central (1), la table de passage de l'espace dépendant de représentation des couleurs du scanner (4) vers un espace indépendant de représentation et la table de passage d'un autre espace indépendant de représentation vers l'espace dépendant de représentation des couleurs de l'imprimante (5) et, dans la mémoire de travail dédiée (7) du microprocesseur secondaire couleur (6), toutes les autres tables de changement d'espace couleur.

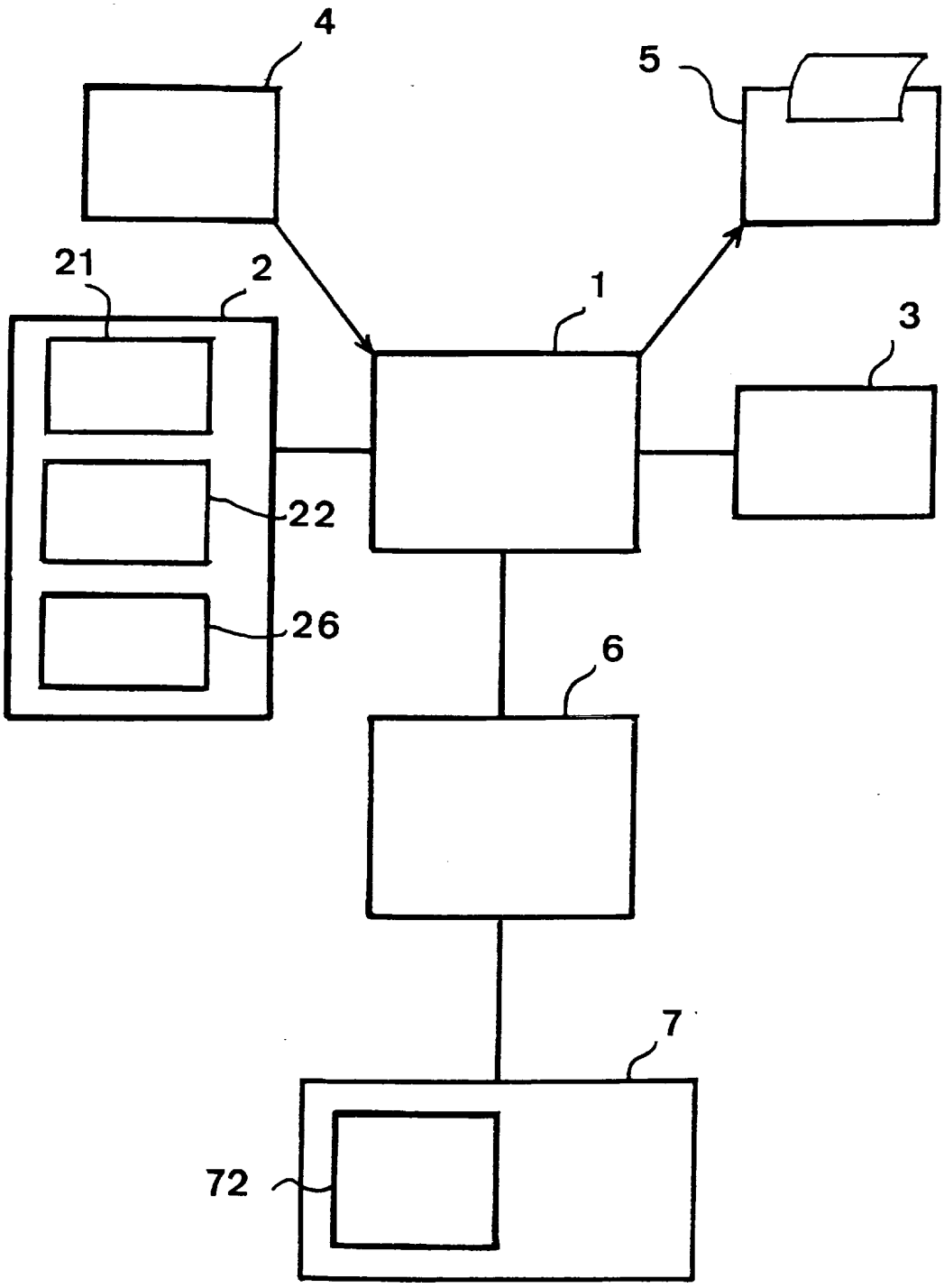


FIGURE UNIQUE

INPI
INSTITUT NATIONAL DE LA PROPRIETE INDUSTRIELLE

RAPPORT DE RECHERCHE PRÉLIMINAIRE

établi sur la base des dernières revendications déposées avant le commencement de la recherche

2803156

N° d'enregistrement national

FA 584458
FR 9916250

DOCUMENTS CONSIDÉRÉS COMME PERTINENTS		Revendication(s) concernée(s)	Classement attribué à l'invention par l'INPI
Catégorie	Citation du document avec indication, en cas de besoin, des parties pertinentes		
X	US 5 489 921 A (KRABBENHÖFT ET AL.) 6 février 1996 (1996-02-06) * colonne 3, ligne 42 - ligne 48; figures 3-5 * * colonne 4, ligne 27 - ligne 65 * * colonne 5, ligne 4 - ligne 37 *	1,2	H04N1/00 H04N1/64 H04N1/46
A	EP 0 534 871 A (EASTMAN KODAK CO) 31 mars 1993 (1993-03-31) * colonne 13, ligne 13 - ligne 58 * * colonne 14, ligne 36 - ligne 58 *	1,2	
A	EP 0 703 701 A (EASTMAN KODAK CO) 27 mars 1996 (1996-03-27)		
A	EP 0 581 590 A (CANON KK) 2 février 1994 (1994-02-02)		
A	WO 94 24626 A (DATA TRANSLATION INC) 27 octobre 1994 (1994-10-27)		
			DOMAINES TECHNIQUES RECHERCHÉS (Int.Cl.7)
			H04N

Date d'achèvement de la recherche: 30 août 2000

Examinateur: Kassow, H

CATÉGORIE DES DOCUMENTS CITÉS

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 bénéficiant d'une date antérieure à la date de dépôt et qui n'a été publié qu'à cette date de dépôt ou qu'à une date postérieure.
D : cité dans la demande
L : cité pour d'autres raisons

& : membre de la même famille, document correspondant

EPO FORM 1503 12.99 (P04C14)