

Maîtrise EEA - EC 232**TRAVAUX PRATIQUES
DE
TRAITEMENT D'IMAGES*****Objectif des manipulations***

Les manipulations de traitement d'images abordent plusieurs aspects de cette discipline :

- les aspects matériels de la prise de vue par caméra d'une scène réelle
- l'évaluation de la qualité du signal "image" et ses propriétés
- les traitements en vue de l'amélioration de l'image ou de son analyse

Les deux manipulations se font sur un logiciel dédié à cette discipline. Bien que ce logiciel ne soit pas un produit commercial, il présente des caractéristiques similaires à de nombreux autres logiciels de ce domaine; les travaux réalisés pourront être facilement transposés sur Matlab par exemple, qui tend à se généraliser de nos jours à de nombreux domaines scientifiques et industriels.

La rédaction de ce polycopié alterne les présentations avec les travaux à réaliser, de façon à mettre en application directement les concepts énoncés. La préparation des manipulations consiste donc à lire avec soin le texte avant la manipulation et à confronter les problèmes posés avec le Cours et les Travaux Dirigés.

Compte-rendu

Les résultats des manipulations seront présentés en traitement de texte (Wordpad) , en utilisant le plus souvent possible la fonction "copier-coller" sous Windows. Pour assurer une bonne lisibilité au compte-rendu, il est souhaitable de limiter le volume du document à une dizaine de pages, en réduisant fortement la taille des images (**une page peut contenir 8 images !**).

Etant donné les risques usuels de "plantage" dur ordinateur, il est impératif de sauver périodiquement les travaux (macros Classman et compte-rendu). Le compte-rendu sera sauvé sur le répertoire personnel de l'étudiant sur le réseau.

Important: le compte-rendu ne se limite pas à un simple "collage" des programmes et travaux réalisés. Le compte-rendu donnera toutes justifications scientifiques aux résultats. **Tout résultat non-justifié sera considéré comme nul.**

Manipulation n° 1

PRISE EN MAIN et ANALYSE DU "SIGNAL IMAGE"

I - PRESENTATION DU MATERIEL

1) La caméra

La caméra est un modèle industriel possédant un capteur du type CCD (Charge Coupled Device ou Dispositif à Transfert de Charges) **Noir&Blanc**, de dimension 1/3 pouce, muni d'un objectif à focale variable, pour adapter le grandissement.

Cet objectif n'étant pas un véritable zoom; les réglages 'focale' et 'distance' ont une influence mutuelle; le réglage optique se fait par action *conjointe* sur les deux bagues.



Figure 1 – Caméra utilisée en manipulation

Le signal fourni par la caméra est directement *proportionnel* à l'énergie reçue par le capteur, sans correction automatique de gain type CAG (la fonction est désactivée lorsque le switch "Auto-iris" de la caméra est sur la position "ON")

Le contrôle de l'amplitude du signal se fait par le diaphragme de l'objectif, bague la plus proche du corps de la caméra. L'ouverture sera ajustée pour obtenir une image *non-saturée* mais suffisamment lumineuse (sauf indication contraire dans le texte).

2) La carte d'acquisition

La carte d'acquisition prend en charge le signal vidéo composite issu de la caméra pour en extraire les signaux logiques de synchronisation et le signal de luminance Y (ou niveau de gris) . Ce signal est échantillonné selon la norme CCIR (576x768 pixels) puis converti sur 256 niveaux (convertisseur 8 bits) . Le couplage vers la mémoire de l'ordinateur se fait par accès direct en mémoire (DMA) au travers du bus PCI dont le débit est de 33Mhz sur 32 bits, ce qui suffit pour faire une acquisition continue d'images couleur en format CCIR

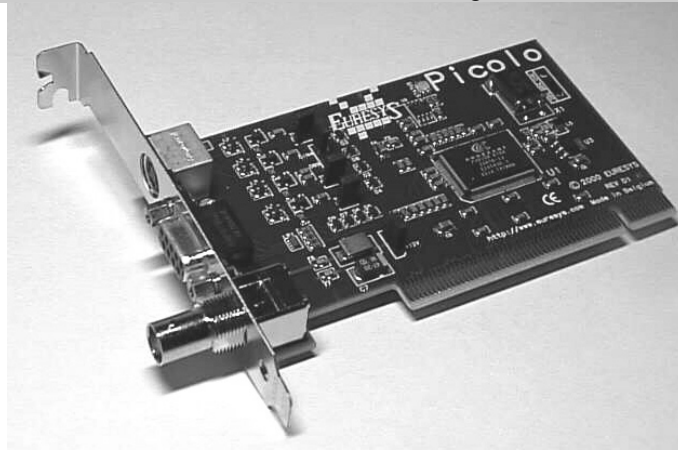


Figure 2 – Carte d'acquisition vidéo PicoLo

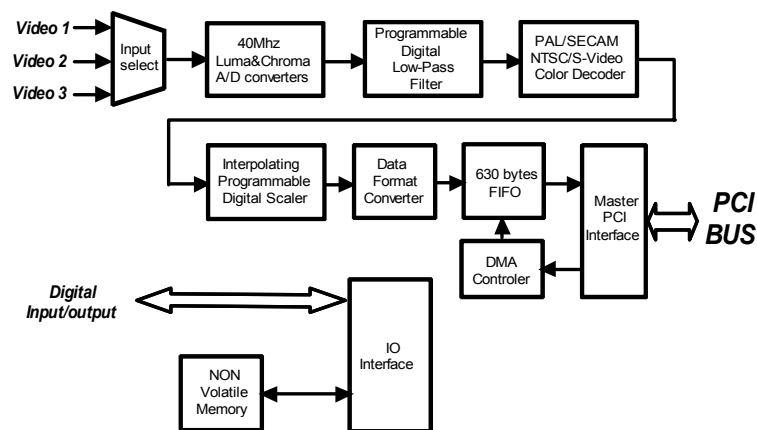


Figure 3 – Schéma fonctionnel de la carte d'acquisition vidéo PicoLo

Cette carte supporte tous les formats usuels de caméra de télévision, y compris les caméras couleur. La configuration de la carte se fait par logiciel, au travers du programme d'acquisition.

3) Le logiciel d'acquisition

Le logiciel d'acquisition se dénomme EasyGrab. Il permet de piloter plusieurs caméras connectées à la carte (jusqu'à 3).

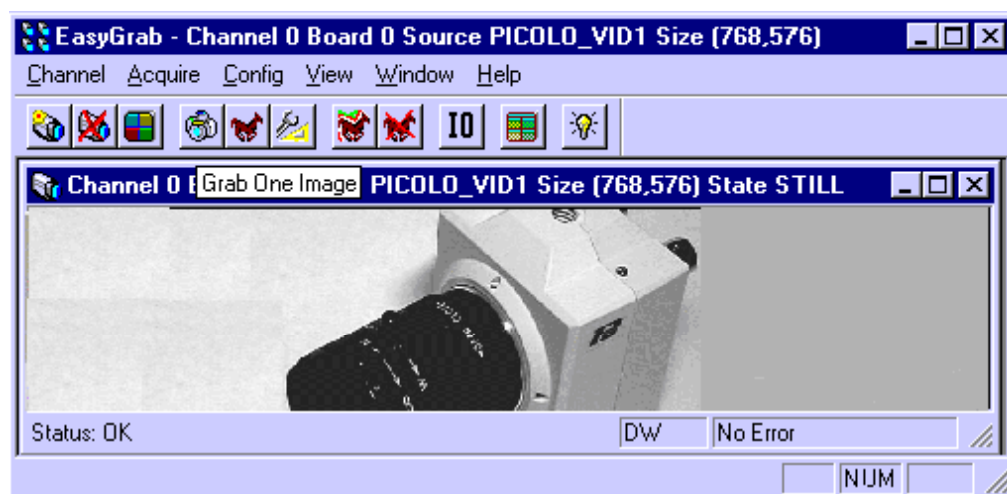




Figure 4 – Menu du logiciel EasyGrab pour PicoLo

A chaque entrée vidéo correspond une voie d'acquisition fonctionnelle ou "channel", possédant ses propres caractéristiques. La configuration par défaut d'une voie est proche des besoins usuels: elle permet de prendre une image en mode CCIR (format européen) sur l'entrée vidéo 1. La caméra étant Noir&Blanc, il est inutile de faire l'acquisition en mode multicanaux.


Travail à faire

- Sélectionner la fonction Channel  et paramétrer l'acquisition pour une caméra Noir&Blanc sur l'entrée 1. Utiliser le mode mémoire Y8, soit l'acquisition de la luminance seule sur 8 bits (256 niveaux de gris).

- Mettre l'acquisition en mode continu 
- Régler la caméra pour obtenir une image de bonne qualité de Lena fournie en annexe.

- Geler l'image par 
- Enregistrer l'image en format TIF dans un répertoire créé sur votre compte (*il est strictement interdit de stocker les images dans un répertoire du disque C des machines autres que "Image"*)

Cette image servira de source pour la prise en main de Classman

- Parcourir les fonctionnalités des autres menus de EasyGrab, en particulier le menu de configuration  et en donner une rapide description fonctionnelle.

II - PRESENTATION DU LOGICIEL CLASSMAN32

1) Fonctionnalités de base

CLASSMAN est un logiciel de visualisation et de traitement d'image qui s'exécute sous l'environnement WINDOWS 95 ou NT.

L'intérêt de CLASSMAN est de proposer des traitements sur les images, issus d'une bibliothèque de fonctions extensible par l'utilisateur (la création de nouvelles fonctions ne fait pas partie de cette série de manipulations).

Comme beaucoup de logiciels du domaine "image", CLASSMAN permet de charger des images, de les observer, de les sauver sous différents formats. Ces fonctions sont directement accessibles par Menu.

Contrairement à de nombreux logiciels grand public dans lesquels les fonctions ne sont accessibles que par menu et ne permettent donc pas de traitement automatisé, CLASSMAN effectue les traitements *obligatoirement* à partir d'un fichier (ou fenêtre) de **Macro-Commandes** formant une suite de commandes élémentaires. CLASSMAN permet l'écriture de programmes complets, en autorisant les boucles (test, sauts...).

Bien que la syntaxe soit "C-like", avec déclaration de variables et réservation d'espace mémoire, le langage de macro-commande n'est pas compilé, ce qui pénalise légèrement les applications en temps d'exécution (ce phénomène est peu sensible en Traitement d'Images pour lequel le temps d'exécution des fonctions est très important devant le temps d'interprétation des commandes)

Il dispose des commandes de mise au point (exécution pas à pas, exécution cyclique...) comme un langage classique..

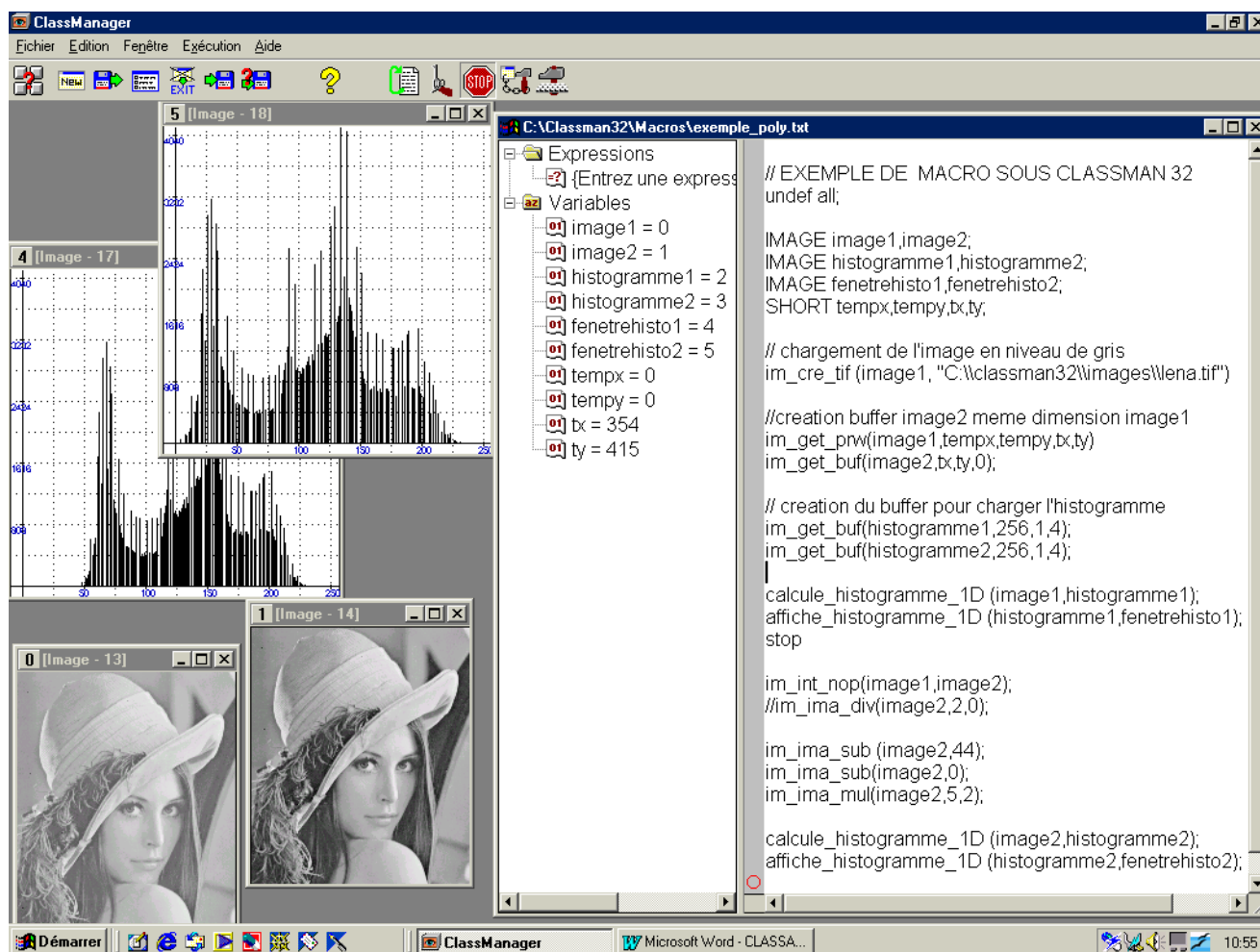


Figure 5 - Ecran typique sous CLASSMAN (correction de dynamique de l'image Lena)

Classman32 affiche deux types de fenêtres:

- les fenêtres images (N&B, couleur, graphique) de taille maximale 1024x1024, avec fonctions (accessibles par menu direct) de Zoom, déplacement, copier/coller, profil, affichage 3D, pointage des pixels (résultat affiché en bas de la fenêtre)
- la (ou les) fenêtre de programmation sous formes de Macro-Commandes interprétées. Cette fenêtre possède sur sa partie gauche un "navigateur" de variables (augmenter sa taille par redimensionnement à la souris pour le faire apparaître)

2) Visualisation d'une image

Pour vous habituer à la visualisation des images, effectuer l'exercice suivant:

Travail à effectuer

- charger l'image LENA précédemment sauvegardée par le menu **Fichier Ouvrir**.
- pour obtenir l'effet de zoom de l'image (fonction par défaut), cliquer sur **Mode Zoom**. A chaque fois que vous cliquez le bouton gauche (droit) de la souris, vous zoomez (dézoomez) l'image.
- pour afficher la courbe représentant les niveaux de gris de la ligne de l'image pointée par la souris (bouton gauche maintenu appuyé), cliquer sur **Mode Profil**.


- la vue 3D correspond au graphe de la fonction bidimensionnelle de niveau de gris. En maintenant la souris enfoncée, le *point de vue* est modifiable.
- le niveau de gris du pixel où se trouve la souris est affiché par la barre de status en bas de l'écran.. Le niveau de gris est affiché pour des images Non-signées, Signées ou couleur **RVB** . Remarquez que la valeur du niveau de gris d'un pixel (comprise entre 0 et 255) représente son intensité. Plus le "pixel est clair", plus son niveau de gris est élevé.
- le mode Sélection définit graphiquement une zone qui pourra être copiée dans le presse-papier (bien que la visualisation ne le montre pas, la zone reste sélectionnée après le lâcher du bouton de souris).

3) Les macro-commandes

La description exhaustive des macro-commandes serait fastidieuse. Nous étudions ici les principes de base; les commandes spécifiques à une application seront introduites au moment opportun.

Écriture d'un programme de macro-commandes :

La fenêtre de macro-commande est un éditeur texte classique dans laquelle les fonctions "copier coller" sont opérationnelles. La saisie d'un programme Classman se fait donc directement dans la fenêtre du logiciel.

Le mode "édition" de la fenêtre est obtenu par le bouton  relâché.

Pour faciliter l'écriture, il existe un navigateur de fonction avec aide au collage et aide contextuelle. Cette facilité permet d'éviter toute faute de syntaxe et toute écriture d'une fonction inexistante.

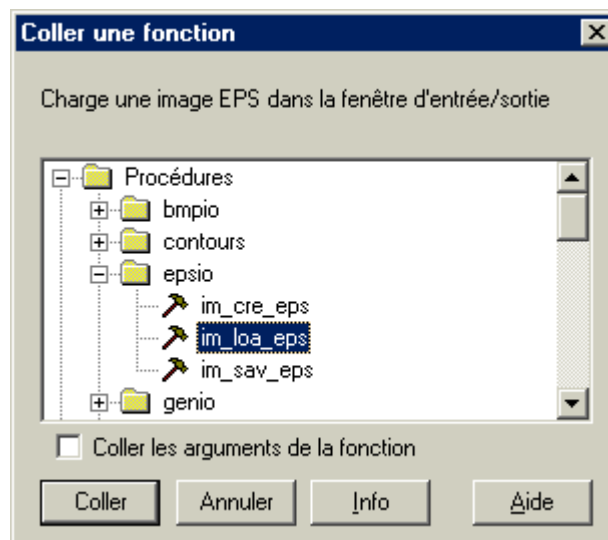






Figure 6 - Navigateur de fonction de Classman

L'option de collage des arguments de la fonction propose de placer des noms standards d'image ou de variable, qu'il faut remplacer par les noms que vous avez adopté dans le programme. Une aide en ligne contextuelle est fournie par le logiciel : il suffit d'appuyer sur F1 pour obtenir l'aide sur la fonction sur laquelle se trouve le curseur d'édition.

Exécution d'un programme

- Pour exécuter un programme, le texte doit être préalablement chargé dans une fenêtre macro et cette fenêtre doit être active.
- Placer le pointeur d'exécution au début de programme par action sur 
- Basculer la fenêtre "Macro" en enfonçant le bouton  en mode "exécution", puis frapper sur la touche "Return" pour exécuter pas à pas le programme.
- L'exécution automatique se fait par appui sur 

Une erreur provoque l'arrêt de l'exécution, le point rouge signalant la position de la ligne en défaut d'interprétation. Il est possible de corriger la faute (erreur de frappe par exemple) en mode "édition" (bouton  relâché) puis de reprendre l'exécution au point d'erreur sans revenir au point de départ.

Les variables Classman

Une variable utilisée dans le programme **doit** être déclarée avant usage. Une différence fondamentale est faite entre les types classiques et le type image : la déclaration d'une image déclare son pointeur mais ne fait pas l'allocation mémoire (faite par une instruction spécifique).

Types numériques:

CHAR nom_variable;	variable 8 bits signée (notation en complément en 2)
UCHAR nom_variable;	variable 8 bits non signée
SHORT nom_variable;	variable 16 bits signée
USHORT nom_variable;	variable 16 bits non signée
LONG nom_variable;	variable 32 bits signée
ULONG nom_variable;	variable 32 bits non signée
DOUBLE nom_variable;	variable réelle

Ces variables peuvent être indicées pour former une liste de valeurs ou un tableau.

exemple : **UCHAR tableau[3][3]** ; **SHORT liste[10]** ;

Types Image

La déclaration d'une variable est :

IMAGE Image1 ;

Le type d'image est défini lors de la demande **d'allocation mémoire** par l'instruction :

im_get_buf(Image1, tailleX, tailleY, n°_type) ;

La réservation mémoire crée automatiquement la fenêtre d'affichage correspondante. Le numéro de fenêtre est donné dans le navigateur de variable.

type 0 : *image 8 bits à niveaux de gris* (non signé par défaut). Attention, certains opérateurs (soustraction, complément, convolution nxn ...) renvoient un format signé. L'affichage ne prend pas en compte le signe (affichage brut des images signées); la valeur définie par le pointeur souris permet de connaître la valeur dans chacun des modes.

type 1 : *image indexée*, format utilisé dans les stockages type TIF (les valeurs des pixels correspondent à un code de couleur stocké dans une table)

type 2 : *image 16 bits*; ce format est utilisé pour des calculs intermédiaires débordant du format 8 bits (addition de deux images 8 bits par exemple). L'affichage tient compte du format (seuls les 8 bits de poids forts sont affichés en réalité).

type 3 : *image couleur 3x8 bits* . L'affichage est directement en couleur. Les images composantes accessibles par "éclatement" de l'image RVB en 3 images indépendantes.

type 4: *image du type long 32 bits* (calculs spécifiques comme l'*histogramme*)

Instruction de chargement image

Classman permet de charger ou de sauver toute variable sur le disque de la machine. Les instructions les plus utilisées sont celles concernant les images.

Tous les formats actuels d'image sont lisibles. Classman peut donc servir de convertisseur de format de stockage !

Exemple de programme de base:

```
undef all; // supprime les anciennes définitions (mémorisées dans l'interpréteur)
IMAGE Image1;
im_get_buf (Image1, 200, 150, 0);
im_loa_tif (Image1,"c:\\image\\lena.tif");
im_sav_eps (Image1, "c:\\image\\lena.eps");
```

Dans cet exemple, l'image est incorrectement chargée, car le buffer est de taille insuffisante pour contenir l'image. Pour faciliter la lecture des fichiers, il existe une instruction pour créer un buffer de la taille convenable pour l'image et y charger le fichier:

```
im_cre_tif (Image1,"c:\\image\\lena.tif");
```

Travail à effectuer

- Ecrire un programme pour charger l'image source LENA précédemment sauvegardée en mode TIF.
- Déterminer la taille de l'image par `im_get_prw (image, origineX,origineY,tailleX,TailleY)`
- Créer un nouveau buffer de même taille et recopier l'image source dans une image 16 bits par `im_int_cnv()`
- Sauvegarder cette image résultat en mode JPEG et comparer la taille des fichiers.
- Reprendre l'expérience avec l'image d'un texte Noir&Blanc (page du polycopié par exemple)

instructions de contrôle de déroulement du programme

Le logiciel Classman possède l'instruction de test dont la syntaxe est :

```
if (test) instruction ;
```

L'instruction de saut est :

```
goto label ;
```

L'étiquette ou label est définie par:

```
label nom_label
```


Exemple d'application : mise à zéro de tous les pixels d'une image (calcul interprété très lent) (les déclarations ne sont pas présentées dans cet exemple) :

```
// pour effectuer une RAZ de l'image
i= 0;
label bouclei
j= 0;
label bouclej
ecrire_niveau_de_gris (image,i,j,0);
j = j+1;
if (j <ty) goto bouclej
i = i+1;
if (i <tx) goto bouclei
```

III – MANIPULATION D'HISTOGRAMMES

1) Histogramme sous Classman

Dans le logiciel Classman, un histogramme est une **image** de dimension **256x1** de type **4** (entiers longs32 bits). Pour visualiser l'histogramme, le logiciel possède une fonction spécifique.

La déclaration de l'histogramme est donc celui d'une image. Pour faciliter la manipulation des histogrammes, la déclaration de l'histogramme fait l'allocation mémoire adapté à ce type particulier :

```
IMAGE nom_histogramme;
im_cre_histogramme_1D (nom_histogramme);
```

Le graphique est tracé dans une fenêtre de type IMAGE de dimension appropriée déclarée par :

```
IMAGE nom_fenetre_histogramme;
im_cre_affiche_histogramme (nom_fenetre_histogramme);
```

Les fonctions de calcul et d'affichage sont :

```
calcule_histogramme_1D (image, nom_histogramme);
affiche_histogramme_1D (histogramme, nom_fenetre_histogramme);
```

Travail à effectuer

- Prendre l'image d'un texte Noir&Blanc
- Afficher l'histogramme du texte.
- Conclure sur l'histogramme du texte (forme, densité, min, max...)
- Reprendre l'expérience avec l'image de Lena et comparer les histogrammes d'une image de type binaire avec celui d'une image "naturelle"
- Reprendre l'image du texte en faisant une prise de vue volontairement trop sombre (fermer le diaphragme de l'objectif) puis une vue trop claire.

2) Egalisation d'histogramme

L'égalisation d'histogramme se fait en modifiant les niveaux de gris d'une image source par une loi de correspondance qui n'est rien d'autre que l'histogramme cumulé normalisé (voir poly de cours).

Cette loi est tabulée pour chaque niveau de gris source; elle prend le nom de **LUT (Look Up Table)** . Dans Classman, les **LUT** sont du type **IMAGE**, de dimension **256x1** , **8bits** (ne pas oublier de déclarer le nom et de faire l'allocation mémoire)

Le calcul de la loi d'égalisation se fait par:

transforme_histogrammecumule_lut (histogramme_cumule, lut);

L'application de la loi de correction aux niveaux de gris est obtenue par:

applique_lut (image, lut, image_egalisee);

Travail à effectuer

- Prendre l'image de Lena
- Afficher son histogramme
- Calculer l'histogramme cumulé par :
transforme_histogramme_histogrammecumule (histogramme, histogramme_cumule);
- Afficher l'histogramme cumulé.
- Appliquer l'égalisation d'histogramme à l'image Lena
- Calculer l'histogramme de l'image égalisée, son histogramme cumulé et l'afficher
- L'opération a-t-elle permis d'obtenir un histogramme plat ? (commenter les résultats en vous servant du polycopié de cours)

3) Calcul direct de l'égalisation

Remarque: ce travail étant assez complexe, il est vivement conseillé de ne pas y consacrer un temps excessif et de passer directement à la suite en cas de difficulté.

On se propose de traiter le problème de l'égalisation sans faire appel à la fonction intégrée au logiciel. Cette technique offre la possibilité de nouvelles applications comme l'obtention d'un histogramme de forme quelconque (non demandé dans cette manipulation). Le calcul comprend trois parties:

a) calcul de l'histogramme cumulé à l'aide de l'instruction suivante utilisée dans une boucle de 0 à 255 :

lire_valeur_histogramme_1D (histogramme, rang_histo,valeur_histo);

La valeur lue de l'histogramme est du type LONG (entier 32 bits).Les valeurs seront accumulées dans une variable de sommation de type LONG ou DOUBLE

La valeur accumulée est affectée à l'histogramme cumulé par:

ecrire_valeur_histogramme_1D (histogramme, rang_histo, valeur, erreur);

Dans cette instruction, la valeur d'erreur (du type USHORT) est nulle si l'instruction a été correctement exécutée.

b) Normalisation de l'histogramme cumulé

Pour obtenir une loi de correction des niveaux de gris, il est indispensable que la valeur finale de l'histogramme cumulé soit 255 pour le niveau de gris 255

Le calcul du coefficient de normalisation se fait aisément sous Classman, puisque la valeur finale de l'histogramme cumulé est égale au nombre total de pixel de l'image obtenu par le calcul suivant défini en type **ULONG** :

nombre_total_pixels = TailleX * tailleY;

Le coefficient de normalisation est donc:

coef_norme = 255 / nombre_total_pixels;

c) Application de la loi aux niveaux de gris

La méthode est la même que précédemment.

Travail à faire

- Modifier le programme précédent, en calculant l'histogramme cumulé par méthode directe
- Afficher l'histogramme cumulé non normalisé
- Calculer l'histogramme cumulé normalisé et l'afficher
- Réaliser l'égalisation
- Conclure

IV – ETUDE DU BRUIT DE QUANTIFICATION

Le bruit de quantification apparaît pour une quantification avec peu de niveaux. L'image 8 bits est prise comme image de référence, en supposant que le bruit de quantification est négligeable pour cette résolution. Pour mettre en évidence ce bruit, on propose la manipulation suivante:

Travail à faire

- Faire l'acquisition de l'image Lena en veillant à utiliser toute la dynamique (le vérifier en traçant l'histogramme de l'image source, envisager une égalisation d'histogramme).
- Par l'instruction de division des niveaux de gris **im_ima_idi()**, réduire le nombre de niveau à 16, puis ramener l'image à la dynamique 0-255 par **im_ima_mul()**
- Par soustraction des deux images, calculer l'image de bruit de quantification et afficher son histogramme. Commenter la forme de l'histogramme
- Calculer moyenne et écart-type de l'image de bruit par la fonction appropriée de la bibliothèque **Contours**.
- En reprenant cette méthode pour une quantification à 2, 4 ... 128 niveaux, tracer la courbe du bruit de quantification (exprimé en dB) en fonction du nombre de niveau de quantification.
- Conclure sur les limites de cette méthode.

Manipulation n° 2

APPLICATIONS DE LA CONVOLUTION

I. LA CONVOLUTION DISCRETE SOUS CLASSMAN

1) Fonctions prédéfinies

Classman possède une bibliothèque de fonctions de convolution prédéfinies pour un voisinage 3x3 . Ces fonctions sont adaptées à un emploi simplifié: le masque est prédéfini et le résultat est pris en valeur absolue, ce qui évite les problèmes de gestion de signe.

La bibliothèque a pour nom **im_c3x3**. Parmi les convolutions disponibles, on retrouve le moyenneur (**im_3x3_ave**), les gradients (modules) de Sobel et Prewitt, ainsi que le Laplacien.

2) Convolution mxn

Pour appliquer une convolution de taille quelconque, il faut en définir le masque. La structure adoptée sur Classman32 est celle d'un tableau d'entiers signés de type **SHORT** (pas type fractionnaire ou réel !)

Exemple de définition de masque:

```
SHORT voisinage_x, voisinage_y;
SHORT masque1[7][7];
SHORT norme_masque1;
voisinage_x = 5;
voisinage_y = 7 ;
masque1[0][0] = x ;
masque1[0][1] = x ;
...
masque[6][6] = x ;
norme_masque1 = xx ;
```

Le coefficient de normalisation est appliqué en division sur le résultat des calculs.

im_nxn_con (image_source, masque,voisinage_x , voisinage_y, norme_masque1, image_ résultat);

Le résultat d'une convolution pouvant être signé localement si le masque comporte des valeurs négatives, il faudra prendre toutes les précautions nécessaires pour ne pas déborder des valeurs [-128, +127] . La première méthode est de diviser le niveau de gris de l'image source par 2 pour limiter sa dynamique à [0,127] . La deuxième méthode est de prendre comme norme du masque la somme des valeurs absolues de ses coefficients.

Dans les deux méthodes, le calcul pratique peut différer de l'approche théorique (cas des masques de dérivation).

II – FILTRES DE LISSAGE ET RESTAURATEURS

1) Lissage itératif d'une image

Les images peuvent comporter un bruit d'acquisition important lié à la technologie de certains capteurs (caméras CMOS par exemple) ou suite à une opération de forte compression (mosaïque spatiale en compression JPEG).

L'objectif est d'effectuer une convolution afin d'atténuer "le bruit" et les irrégularités d'une image.

Pour tester la convolution 3x3, on utilise le masque lisseur MOY3x3 suivant :

1	1	1
1	1	1
1	1	1

avec comme facteur de normalisation **1/9**.

Travail à faire

- Acquérir l'image d'une page de texte (image binaire)
- Ajouter un bruit gaussien par **im_gen_gss()** (bibliothèque contours) d'écart-type 10
- Sauvegarder cette image sous le nom **TEXTEBRUIT.tif**
- Lisser l'image avec **MOY3x3** et stocker le résultat dans **MOY3X3.tif**
- Lisser l'image une deuxième fois avec **MOY3x3** et stocker le résultat dans **MOY5X5.tif**
- Comparer les images lissées avec un passage puis deux passages du moyeneur 3x3. Quelle est l'influence des passages successifs de la convolution sur le bruit et l'aspect de l'image?

2) Convolution directe 5x5

On définit le masque de lissage 5x5 suivant

1	2	3	2	1
2	4	6	4	2
3	6	9	6	3
2	4	6	4	2
1	2	3	2	1

le facteur de normalisation étant 1/81.

Travail à effectuer

- Charger l'image **TEXTEBRUIT.tif**
- Lisser l'image avec le lisseur 5x5 défini ci-dessus et stocker le résultat dans **RES5X5.tif**
- Comparer visuellement les images **MOY5X5** et **RES5X5**.
- Faire la comparaison par soustraction des deux images
- Conclusion et justification des résultats

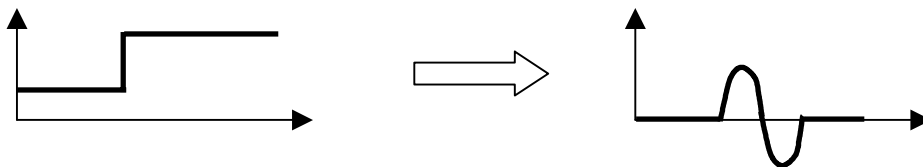
3) amélioration du contraste par accentuation des contours.

Cette opération est l'inverse d'un lissage car elle consiste en une amplification des hautes fréquences spatiales qui met en relief les transitions rapides. Une image peut être rehaussée par application d'un opérateur permettant de détecter les contours : le résultat additionné à l'image initiale permettra alors d'obtenir une image rehaussée. Cette opération peut s'effectuer, par exemple, en appliquant l'opérateur Laplacien :

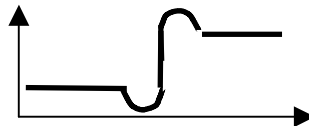
$$\begin{array}{|c|c|c|} \hline 0 & -1 & 0 \\ \hline -1 & 5 & -1 \\ \hline 0 & -1 & 0 \\ \hline \end{array} = \begin{array}{|c|c|c|} \hline 0 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 0 \\ \hline \end{array} - \begin{array}{|c|c|c|} \hline 0 & 1 & 0 \\ \hline 1 & -4 & 1 \\ \hline 0 & 1 & 0 \\ \hline \end{array}$$

Rehausseur *Elt Neutre* *Laplacien*

Remarque : pour une transition franche de type marche (échelon), l'opérateur Laplacien donne ceci :



Donc pour le résultat retranché à l'image originale, on obtient effectivement une image améliorée par accentuation des contours :



Travail à faire

- Créer le masque **REHAUSS** avec comme facteur de normalisation 1.:
- Lisser l'image LENA avec le masque **MOY3X3**
- Restaurer les contours avec **REHAUSS**. Conclusions .
- Reprendre le travail avec l'image **TEXTEBRUIT.tif** .Que se passe-t-il sur l'image résultat? A quels problèmes de calcul sont dus les éventuels mauvais résultats?
- Appliquer ce filtre sur une image rendue floue par défaut de mise au point. Peut-on appliquer plusieurs fois successivement ce filtre pour obtenir une "bonne" image ?

III - FILTRES DE DERIVATION

1) Dérivation dans les directions de la maille d'échantillonnage

Il arrive souvent qu'on cherche à mettre en évidence les transitions dans une direction particulière. Par exemple, l'opérateur $\frac{\partial f(x,y)}{\partial x}$ met en évidence les contours verticaux et a tendance à gommer les contours horizontaux. Inversement $\frac{\partial f(x,y)}{\partial y}$ fait ressortir les contours horizontaux et a tendance à gommer les contours verticaux.

-1	0	1
-2	0	2
-1	0	1

$$H_1 = \frac{\partial f(x,y)}{\partial x}$$

-1	-2	-1
0	0	0
1	2	1

$$H_2 = \frac{\partial f(x,y)}{\partial y}$$

Les résultats d'une dérivation étant signés, les images résultat demandent une interprétation. Pour ce type d'image, il est proposé de décaler les niveaux de gris:

- le zéro est translaté en 128 soit un gris moyen
- les valeurs négatives sont traduites par des niveaux plus sombres
- les valeurs positives sont traduites par des niveau plus clairs

Travail à effectuer

- Créer les masques **H₁** et **H₂**
- Acquérir l'image TEXTE sans bruit.
- Appliquer les deux filtres sur cette image . Commenter les valeurs brutes obtenues (voir les niveaux de gris dans la fenêtre inférieure). Au besoin, tracer l'histogramme de l'un des résultats pour expliciter les valeurs obtenues.
- Proposer une visualisation des grandeurs signées par décalage des niveaux de gris (utiliser la fonctions **im_ima_add()**)
- Reprendre le travail avec l'image **TEXTEBRUIT.tif** .
- Conclure sur l'efficacité des dérivateurs usuels

2) Dérivée dans une direction quelconque

Il peut être intéressant de chercher à mettre en valeur les contours dans d'autres directions en utilisant l'opérateur $\frac{df(x,y)}{d\alpha}$. Cet opérateur peut s'obtenir de la façon suivante :

$$\frac{df(x,y)}{d\alpha} = \frac{\partial f}{\partial x} \cos \alpha + \frac{\partial f}{\partial y} \sin \alpha$$

Travail à effectuer

- Repartir du programme précédent donnant les dérivées dans les directions horizontales et verticales pour calculer la dérivée dans la direction $\alpha = \frac{\pi}{4}$ (utiliser les fonctions de multiplication/division `im_ima_imu()` du niveau de gris par un entier et d'addition du logiciel Classman)
- Visualiser l'image résultat
- Donner la méthode de pour une direction quelconque

IV - SEGMENTATION D'UNE IMAGE BRUITEE

La segmentation d'une image consiste à isoler les différents objets présents dans une image.

La méthode la plus simple de segmentation d'une image est de *seuiller* ou *binariser* une image à partir des niveaux de gris des pixels.

Soit S une valeur de seuil et $f(x,y)$ le niveau de gris d'un pixel de coordonnées x et y . Supposons que l'image représente un objet clair (niveaux de gris élevé) sur un fond sombre (niveaux de gris bas). La règle de binarisation est :

**si $f(x,y) < S$
alors le pixel $(x,y) \in$ fond de l'image; sinon le pixel $(x,y) \in$ objet.**

Travail à faire

- Charger l'image TEXTEBRUIT.tif
- Calculer et afficher l'histogramme de l'image et en déduire une valeur de seuil de binarisation.
- Binariser l'image et décrire le résultat obtenu
- Ajouter un bruit gaussien d'amplitude 10 à l'image source
- Binariser cette image avec la fonction `im_ima_bin()`. Quelle est la qualité du résultat pour ce type d'images ?
- Lisser l'image bruitée avec un moyennneur 3x3 puis binariser . Montrer l'intérêt du lisseur dans ce type d'application.

