

Lynx

Stéphane Poirier

COLLABORATORS

	<i>TITLE :</i> Lynx		
<i>ACTION</i>	<i>NAME</i>	<i>DATE</i>	<i>SIGNATURE</i>
WRITTEN BY	Stéphane Poirier	December 7, 2024	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

1	Lynx	1
1.1	Lynx Version 1.0ß	1
1.2	Conditions d'utilisation	1
1.3	Introduction	2
1.4	Caracteristiques générales	2
1.5	Objectifs	3
1.6	Concepts	4
1.7	Installation	5
1.8	Guide d'utilisation	5
1.9	Menus de l'écran de contrôle	5
1.10	Ouvrir une image au format Lynx	6
1.11	Ouvrir une image au format IFF	7
1.12	Ouvrir une image au format FITS	7
1.13	Ouvrir une image à l'aide des datatypes	7
1.14	Ouvrir une image au format IMA	8
1.15	Ouvrir une image au format PIC	9
1.16	Sauvegarder une image au format Lynx	9
1.17	Sauvegarder une image au format IFF	9
1.18	Sauvegarder une image au format IMA	10
1.19	Sauvegarder une image au format FITS	10
1.20	A propos de Lynx...	10
1.21	Quitter	10
1.22	Définir composantes RVB	11
1.23	Transforme RVB vers TSI	11
1.24	Transforme TSI vers RGB	12
1.25	Calcul Couche Médiane	12
1.26	Statistiques	12
1.27	ARexx	13
1.28	Messages d'erreurs ARexx	13
1.29	Visualisation automatique	14

1.30	Histogramme automatique	14
1.31	Gestion des Images	15
1.32	Buffers d'images	15
1.33	Images monochromes	15
1.34	Piles d'images	16
1.35	Sélection d'une image	16
1.36	Effacer une image	17
1.37	Informations	17
1.38	Modes	18
1.39	Affichage	19
1.40	Choix du mode d'écran	20
1.41	Réglage des seuils	20
1.42	Transcodage	21
1.43	Echelle de visualisation	22
1.44	Tramage	22
1.45	Visualisation	23
1.46	Visualisation d'images monochromes	23
1.47	Visualisation en mode 'pseudo-couleurs'	24
1.48	Visualisation en mode contour	24
1.49	Visualisation en mode mosaïque	25
1.50	Visualisation d'images en trichromie	25
1.51	Mode 'normal'	25
1.52	Mode Hold And Modify	25
1.53	Histogramme	26
1.54	<<	27
1.55	Couche	27
1.56	Sauver	28
1.57	Min Max % Niveau Nombre de points	28
1.58	Menus de l'écran de visualisation	29
1.59	Boîte à outils	30
1.60	Transformations géométriques	32
1.61	Translation	32
1.62	Rotation	33
1.63	Changement d'échelle	33
1.64	Redimensionnement	34
1.65	Rééchantillonnage	34
1.66	Traitements d'images	35
1.67	dans le domaine spatial	35
1.68	Convolution discrète	35

1.69	Convolution par une matrice	36
1.70	Convolution croisée	38
1.71	Filtrage non linéaire	39
1.72	Transcodage d'image	40
1.73	Opérations arithmétiques	42
1.74	MultConstant	42
1.75	AddConstante	42
1.76	Addition	43
1.77	Soustraction	43
1.78	Division	44
1.79	Normalisation	44
1.80	dans le domaine fréquentiel	44
1.81	Transformation de Fourier	45
1.82	Transformée directe	45
1.83	Transformée inverse	45
1.84	Théorie	46
1.85	Convolution	47
1.86	Déconvolution	48
1.87	Corrélation	49
1.88	Convolution par une fonction analytique	50
1.89	LynxFormat	51
1.90	Format IFF-ILBM	51
1.91	FITS	52
1.92	IMA	52
1.93	PIC	52
1.94	Objet (datatypes)	53
1.95	Utilisation d'ARexx	53
1.96	AddImage	54
1.97	SubImage	54
1.98	AddrCurrentImage	55
1.99	SelectImage	56
1.100	Convole	56
1.101	CurrentImageName	57
1.102	DeleteBuffer	58
1.103	Normalize	58
1.104	RotateImage	59
1.105	Mult	60
1.106	Mask	60
1.107	Mask_Status	61

1.108Sigma	62
1.109LynxEnd	62
1.110Sauver Fenêtre	63
1.111Sauver Ecran	63
1.112Composition d'images	63
1.113Méthode	64
1.114Transition	65
1.115Couche suivante	67
1.116Démarrer animation	67
1.117Vitesse d'animation	68
1.118Libère BitMaps	68
1.119HELP !	68
1.120Sortir	69
1.121Editer	69
1.122Charger	69
1.123Sauver	69
1.124Information	70
1.125Définir Masque	70
1.126Extraire	71
1.127Abandonner	71
1.128Histogramme	71
1.129Tout Effacer	71
1.130Générer Image	72
1.131ToolTypes	72
1.132Correction pixel par pixel	73
1.133Fonte...	73
1.134Défaire	74
1.135Effacer tous les graphiques	74
1.136Intégrer dans l'image	74
1.137Graphiques	74
1.138Exemples pratiques	75
1.139Faire une image couleur avec trois images monochromes	75
1.140Faire varier contraste et luminosité d'une image	76
1.141Réaliser un effet du style 'peinture à l'huile'.	76
1.142 Réaliser une solarisation	77
1.143 Mise en orbite autour de Saturne !	78
1.144Eliminer le bruit présent dans l'image	79
1.145Réaliser un 'masque flou'	80
1.146Réaliser une convolution dans le domaine de Fourier	81

1.147Réaliser une déconvolution dans le domaine de Fourier	82
1.148Un peu de reconnaissance de formes...	83
1.149 Approche de l'analyse en ondelettes	85
1.150Fonction non implémentée	87
1.151Historique	87
1.152A faire	87
1.153Bogues	89
1.154Stéphane Poirier	89
1.155Enregistrement	90
1.156Copyright	90
1.157Remerciements	91
1.158Index	91

Chapter 1

Lynx

1.1 Lynx Version 1.0β

```

##      ##      ##      ##      ##      ##      ##
##      ##      ##      ###     ##      ##      ##
##      ##      ##      #####  ##      ##      ##
##      #####    ##      #####    ###
##      ##      ##      ###     ##      ##      ##
##      ##      ##      ##      ##      ##      ##
#####      ##      ##      ##      ##      ##

```

Version 1.0β

Lynx est Copyright © 1993, 1994, 1995 ~Stéphane~Poirier~

```

~Conditions~d'utilisation~ Ce qu'il faut savoir pour utiliser Lynx
~Introduction~~~~~ Ce que l'on peut faire avec
~Installation~/~Démarrage~ Pour être sur que ca va marcher
>> ~Guide~d'utilisation~~~~~ << Le mode d'emploi...
~Exemples~pratiques~~~~~ Un peu d'apprentissage
~A~faire~dans~le~futur~~~~ Et ce sera génial...
~Bogues~~~~~ Qui seront corrigés
~Copyright~~~~~ Conditions d'utilisation et de diffusion
~Enregistrement~~~~~ Enregistrez-vous !
~Remerciements~~~~~ Encore merci.

```

--

1.2 Conditions d'utilisation

Conditions d'utilisation
=====

Pour utiliser Lynx il faut:

- o Disposer du WorkBench 2.04+.

Wb2.1+ pour accéder à la localisation de Lynx
Wb3.0+ pour accéder à la lecture des datatypes.

- o Disposer d'un disque dur.
- o Avoir au minimum 1Mo de mémoire. Une mémoire de 4 Mo et + est cependant conseillée pour une utilisation optimale.
- o Avoir déjà installé la reqtools.library qui n'est pas livrée avec Lynx.

--

1.3 Introduction

Introduction
=====

I ~Caractéristiques~générales~
II ~Objectifs~
III ~Concepts~
IV ~Historique~

--

1.4 Caractéristiques générales

Caractéristiques générales
=====

Lynx est un logiciel d'analyse et de traitement des images. Il permet de manipuler des images monochromes codées sur des mots de 8 bits non signés, ou 16 bits signés. On peut associer plusieurs images monochromes pour former une image 'PileImages' et traiter ainsi directement des images en 'vraies couleurs'.

On peut résumer ainsi ses caractéristiques:

Capacités:

~~~~~

- o Traitement d'images monocouche ou multicouches (jusqu'à 256 couches).
- o Format d'image allant jusqu'à 4096x2048.
- o Dynamique de 8 bits (0 à 255) ou 16 bits (-32768 à +32767).

Formats supportés:

~~~~~

En ce qui concerne les formats d'images, Lynx n'a pas vocation à lire tous les formats existants sur le marché, ce n'est pas, en effet, un logiciel permettant les conversions entre formats standard.

Les formats supportés sont les suivants:

```
~LynxFormat~
~IFF-ILBM~
~FITS~
~IMA~
~PIC~
~Objet~(datatypes)~
```

Traitements dans le domaine spatial:

~~~~~

- Filtrage par matrice de convolution 3x3 ou 5x5.
- Convolution croisée (gaussienne).
- Filtrage non linéaire.
- Transcodage.
- opérations arithmétiques.

Traitements dans le domaine spectral:

~~~~~

Ces traitements ne seront applicables que sur des couches individuelles d'images.

- Analyse de Fourier: calcul de la Transformée de Fourier directe et inverse en 2D.
- Convolution d'une image avec une autre.
- Convolution d'une image par une fonction analytique.
- Calcul de la corrélation entre deux images.

Manipulations d'images:

~~~~~

- Extraction de partie d'images
- Composition d'images (fondus, mosaïques)
- Rotation, translation, changement d'échelle, redimensionnement, inversion.

Visualisation:

~~~~~

- En Pseudo-couleurs (16 a 256 couleurs avec possibilité de tramages).
- En Vraies couleurs en mode bitplan normal de 4 à 8 bitplans avec possibilité de tramage (méthode de Floyd-Steinberg) ou en mode HAM (6 ou 8).

--

1.5 Objectifs

Objectifs

=====

Son premier objectif est d'améliorer la qualité des images. Ces Images peuvent être issues directement d'une caméra CCD (en astronomie notamment) ou de tout autres sortes d'instruments.

Ses possibilités en matière de traitement sont nombreuses, on peut ainsi:

- o Restaurer une image ayant subit une dégradation (bruit, flou, bougé...).
- o Fabriquer des mosaïques.
- o Analyser les images (valeurs pixel par pixel, histogrammes, coupes photométriques...).
- o Composer en une seule images plusieurs vues du même objet grâce à la grande précision des transformations géométriques.
- o Commander Lynx grâce à son port ARexx (celui-ci n'est pas encore très complet).

Lynx existe en deux versions: une 16 bits et une 8 bits. Pourquoi deux versions allez-vous pensez et pourquoi une version 16 bits ?

En fait la version 16 bits est plutôt réservée aux traitements les plus objectifs (restauration d'images, études scientifiques...) car elle offre une grande dynamique (16 bits par pixel et par canal), comparable à celle utilisée par les logiciels professionnels.

La version 8 bits est plus destinée aux traitements subjectifs (amélioration de la qualité artistique des images) car elle ne travaille qu'avec une dynamique de 8 bits par pixels et par composante (soit 24 bits en couleurs), insuffisante pour un travail de précision.

--

1.6 Concepts

Concepts
=====

Lynx peut stocker en mémoire plusieurs images simultanément; le nombre d'images pouvant être stockées ne dépendant que de la capacité de la machine.

La notion d'image couleur lui est inconnue, il ne connaît que les images monocouches ou multicouches. Le nombre de couches peut aller jusqu'à 256 (limite arbitraire jugée suffisante).

Une image monocouche est équivalente à une image monochrome.

Un Image multicouche peut être:

- Une image en couleur qui sera composée de 3 couches: une pour le rouge, une pour le vert et une pour le bleu, par exemple.
- Une image multispectrale où chaque canal sera représenté par une couche.

- Une image 3D et ici le nom de 'couche' prend tout son sens !

--

1.7 Installation

Installation/Démarrage
=====

Installation
~~~~~

Pour installer Lynx, cliquer sur l'icône d'installation (située dans le répertoire installation) correspondant à la langue que vous désirez.

Le logiciel ainsi que tous les fichiers nécessaires seront correctement installés, et Lynx sera immédiatement après prêt à l'emploi.

Démarrage  
~~~~~

Avant de démarrer Lynx, vous pouvez fixer certains paramètres grâce aux ~ToolTypes~ disponibles avec l'icône d'Image.

--

1.8 Guide d'utilisation

Guide d'utilisation
=====

Chapitre 1: ~Menus~de~l'écran~de~contrôle~
Chapitre 2: ~Gestion~des~Images~
Chapitre 3: ~Modes~de~traitement~
Chapitre 4: ~Affichage~
Chapitre 5: ~Visualisation~
Chapitre 6: ~Transformations~géométriques~
Chapitre 7: ~Traitements~d'images~
Chapitre 8: ~Utilisation~d'ARexx~

--

1.9 Menus de l'écran de contrôle

Chapitre 1: Menus de l'écran de contrôle
=====

Projet
Ouvrir sous...

```

~LynxFormat~
~IFF~image~~~~~
~FITS~image~~~~~
~Objet~~~~~
~IMA~~~~~
~PIC~image~~~~~
Sauver sous...
~LynxFormat~
~IFF~image~~~~~
~IMA~~~~~
~FITS~image~~~~~
-----
~A~propos~de~Lynx...~
-----
~~Quitter~~~~~

```

Pile Active

```

~Définit~composantes~RVB~
Modèle De Couleur
~Transforme~RVB~vers~HSB~
~Transforme~HSB~vers~RVB~
~Calcul~Couche~Médiane~~~

```

Couche Active

```

~Statistiques~

```

Divers

```

~ARexx~~~~~
~Visualisation~automatique~
~Histogramme~automatique~~~

```

--

1.10 Ouvrir une image au format Lynx

```

Ouvrir une image au format Lynx
=====

```

Les deux versions de Lynx peuvent chacune relire les images sauvées par l'autre au format Lynx:

Les images 8 bits seront lues et transformées en 16 bits signés dans Lynx-16.

Les images 16 bits seront lues par Lynx (version 8bits) et les 8 bits de poids les plus fort seront conservés. Les valeurs négatives seront mises à zéro.

Si le fichier .lynx contient plusieurs couches, alors le ~mode~Destination~ est ignoré et une nouvelle pile d'Image est créée pour recevoir l'image ~multicouches~.

Si le fichier .lynx ne contient qu'une ~image~monochrome~, alors elle est

rangée conformément au ~mode~Destination~.

Voir aussi la description du ~format~Lynx~.

--

1.11 Ouvrir une image au format IFF

```
Ouvrir une image au format IFF
=====
```

Lit les formats IFF-ILBM suivants:

IFF de 1 à 8 bitplans
IFF 24 bitplans

2 Possibilités sont offertes:

- a - Charger l'image en couleur, elle sera alors sur 3 couches
- b - Charger et transformer directement l'image en noir et blanc (une seule couche).

Voir aussi le support du format ~IFF-ILBM~.

--

1.12 Ouvrir une image au format FITS

```
Ouvrir une image au format FITS
=====
```

Lit une Image FITS.

Les images FITS sont nécessairement ~monochromes~.

En ~mode~Destination~ 'NouvCouche', cette opération ajoute l'image en queue de liste des couches de l'image active et la redimensionne si nécessaire.

En mode Destination 'NouvPile', cette opération crée une nouvelle pile d'images et y stocke l'image lue.

Voir aussi la description du ~format~FITS~.

--

1.13 Ouvrir une image à l'aide des datatypes

Ouvrir une image par les datatypes

=====

Cette option permet de lire de nombreux formats d'images non reconnus par Lynx. Il faut être au moins sous Workbench 3.0 pour avoir accès à cette possibilité.

Une image lue grâce aux datatypes sera toujours transformée en une image composée de trois couches: une rouge, une verte et une bleue.

Les format suivants (la liste n'est pas exhaustive) sont accessibles ainsi:

GIF
JPEG
PCX
TGA
BMP

Les datatypes correspondants sont faciles à trouver dans le domaine public.

Cependant les datatypes ont une particularité handicapante vis-à-vis de Lynx: Les images sont transformées en bitmaps affichables uniquement, soit 8bits dans le meilleur des cas (Lynx ne sait pas encore lire le HAM8), donc pas de 24bits. On perd ainsi beaucoup en qualité de l'information.

--

1.14 Ouvrir une image au format IMA

Charger une image au format IMA

=====

Les deux versions de Lynx peuvent chacune relire les deux types de codage (8 ou 16 bits) du format IMA.

Les images 8 bits seront lues et transformées en 16 bits signés dans Lynx-16.

Les images 16 bits seront lues par Lynx (version 8bits) et les 8 bits de poids les plus fort seront conservés. Les valeurs négatives seront mises à zéro.

En ~mode~Destination~ 'NouvCouche', l'image lue sera placée comme dernière couche de la pile d'image active ou de la pile d'image à laquelle la couche active appartient.

En ~mode~~Destination~ 'NouvPile', une nouvelle pile d'images sera créée pour y accueillir l'image lue qui sera donc l'unique couche de cette pile.

Voir aussi la description du ~format~IMA~.

--

1.15 Ouvrir une image au format PIC

Charger une image au format PIC
=====

Lit une Image PIC.

Les images PIC sont nécessairement `~monochromes~`. En `~mode~Destination~` `'NouvCouche'` ajoute l'image en queue de couches de l'image active et la redimensionne si nécessaire. En mode Destination `'NouvPile'` crée une nouvelle pile d'images et y stocke l'image lue. Voir aussi la description du `~format~FITS~`.

Voir aussi la description du `~format~PIC~`.

--

1.16 Sauvegarder une image au format Lynx

Sauvegarder une image au format Lynx
=====

Si le `~mode~Source~` est fixé sur `'MonoCouche'`, alors la sauvegarde ne concerne que la couche active

Si le `~mode~Source~` est fixé sur `'PileImages'`, alors la sauvegarde concerne toutes les couches de la pile active.

--

1.17 Sauvegarder une image au format IFF

Sauvegarde au format IFF
=====

Effectue une sauvegarde en IFF 24 bits.

Cette fonction est donc réservée aux images possédant au moins 3 couches, représentant de préférences les canaux rouge, vert, bleu.

L'image est sauvegardée telle qu'elle apparaît lors de la visualisation en trichromie, c'est-à-dire en respectant les seuils et les tables de transcodage.

--

1.18 Sauvegarder une image au format IMA

```
Sauvegarde au format IMA
=====
```

Sauvegarde la couche courante au format IMA.

Suivant la version de Lynx utilisée, la sauvegarde se fera :

- soit en 8 bits non signés,
- soit en 16 bits signés.

--

1.19 Sauvegarder une image au format FITS

```
Sauvegarde au format FITS
=====
```

Sauvegarde la couche courante au format FITS.

Suivant la version de Lynx utilisée, la sauvegarde se fera :

- soit en 8 bits non signés,
- soit en 16 bits signés.

--

1.20 A propos de Lynx...

```
A Propos...
=====
```

Donne des informations sur le programme :

```
Numéro de version
Auteur
...
```

--

1.21 Quitter

```
Quitter
=====
```

Pour quitter Lynx.

Une dernière chance est cependant offerte pour ne pas quitter.

--

1.22 Définit composantes RVB

Définir Composantes RVB
=====

Fonction utilisable uniquement sur une image composée d'au moins trois couches.

Permet de déterminer quelles seront les couches représentant les canaux Rouge, Vert et Bleu pour l'affichage en trichromie.

--

1.23 Transforme RVB vers TSI

Transforme RVB vers TSI
=====

Fonction utilisable sur une image composée d'au moins trois couches.

Cet opérateur permet de changer de système colorimétrique pour une image comprenant au moins les canaux Rouge, Vert et Bleu. (Si les canaux RVB n'ont pas déjà été attribués aux couches de l'image, Lynx propose alors de le faire).

L'opération permet de passer du système de couleur Rouge, Vert, Bleu au système Teinte, Saturation, Intensité (TSI en français, HSV en anglais).

Aucune nouvelle image n'est créée, mais les couches RVB de l'image sont affectées de la façon suivante:

Couche Rouge -> Teinte
Couche Verte -> Saturation
Couche Bleue -> Intensité

La couche Teinte contient l'information COULEUR de chaque pixel (rouge, vert, jaune, violet, bleu... et toutes les nuances possibles),

La couche Saturation contient l'information PURETE de la couleur de chaque pixel:

Un rouge vif est une couleur pure (saturation forte)

Un blanc cassé est une couleur "délavée" (saturation faible)

La couche Intensité contient l'information INTENSITE LUMINEUSE de chaque pixel.

NOTE IMPORTANTE

=====

La transformation RVB->TSI n'a de sens que si les seuils de visualisation de chaque couche RVB sont: 0 pour le seuil min et 255 ou 32767 (suivant la version de Lynx) pour le seuil max.

Si les seuils de visualisation sont différents de cela, les composantes TSI ne reflèteront pas la visualisation de l'image Rouge, Vert Bleu.

--

1.24 Transforme TSI vers RGB

Transforme TSI vers RVB

=====

Effectue la transformation d'une image couleur du système colorimétrique Teinte, Saturation, Intensité vers le système Rouge, Vert, Bleu

Pour avoir plus d'information sur le système TSI, voir la transformation ~RVB~ ~vers~TSI~.

NOTE IMPORTANTE

=====

Il n'y a pas de vérification de la nature des 3 couches, c'est-à-dire que l'on peut effectuer cette opération sur une image couleur déjà représentée par le système Rouge, Vert, Bleu, ce qui n'a pas de sens !

--

1.25 Calcul Couche Médiane

*** Fonction non encore implémentée, désolé !

--

1.26 Statistiques

Statistiques

=====

Calcul les principales grandeurs statistiques de la couche active, à savoir:

Valeur minimale,
Valeur maximale,
Valeur moyenne,

Ecart type.

Il faut noter que l'écart type n'a pas un sens très profond si la distribution des intensités n'est pas de forme gaussienne (pour cela voir l'~histogramme~).

--

1.27 ARexx

Scripts ARexx
=====

Lynx permet d'utiliser ARexx pour effectuer les tâches répétitives ou complexes. Pour cela un port ARexx a été implanté.

L'interface ARexx permet de lancer l'exécution de scripts ARexx.

La liste des scripts ARexx utilisables par Lynx est listée dans le fichier 'Lynx:Scripts'.

La première ligne indique le nombre de scripts et les lignes suivantes contiennent les noms des scripts

Pour le moment l'on doit éditer ce fichier à la main si l'on veut ajouter ou éliminer des scripts ARexx.

Les boutons 'Ajouter' et 'Effacer' restent pour le moment sans effet.

Lynx renvoie différents ~messages~d'erreurs~ selon que le traitement s'est bien ou mal passé:

--

1.28 Messages d'erreurs ARexx

Messages d'erreur ARexx générés par Lynx
=====

Les fonctions de serveurs fournies par Lynx peuvent renvoyer trois types d'erreurs:

- o erreur 1 : Il n'y a aucune image en mémoire.
- o erreur 2 : Les paramètres passés à la fonction ne sont pas correct.
- o erreur 3 : Le processus n'a pas été effectué. Deux causes sont possibles:
 - soit l'utilisateur a sélectionné le bouton 'Abandon'

- d'un processus en cours de réalisation
- soit le processus n'a pu s'effectuer par manque de mémoire.

Dans chacun de ces trois cas le script est interrompu et la main est rendue à Lynx.

--

1.29 Visualisation automatique

Visualisation automatique
=====

Cette option permet de visualiser automatiquement l'image courante après tout traitement effectué.

Le mode de visualisation par défaut est le mode ~Pseudo-couleur~, autrement ce sera le dernier mode de visualisation utilisé (Trichromie, Multi-Images...)

La visualisation automatique sera également activée après :

~Mode~d'Ecran...~
~Transcodage~ (menu AFFICHAGE)
~Histogramme~

--

1.30 Histogramme automatique

Histogramme automatique
=====

L'activation de cette option provoque l'ouverture de la fenêtre d'analyse de l'histogramme après chaque traitement.

Cela est utile pour ajuster rapidement les seuils de visualisation après une opération affectant fortement l'histogramme (convolution avec filtres laplacien, gradient...).

Si l'option de ~Visualisation~automatique~ est également activée, la visualisation aura lieu immédiatement après la fermeture de la fenêtre d'analyse de l'histogramme, sauf dans le cas de la sélection du bouton 'Abandon'.

L'ouverture automatique de l'histogramme sera également activé après :

```
~Mode~d'Ecran~  
~Transcodage~ (menu AFFICHAGE)
```

--

1.31 Gestion des Images

```
Chapitre 2: Gestion des Images  
=====
```

Ce chapitre décrit la manière dont Lynx gère les images présentes en mémoire.

```
2.1 ~Buffers~d'images~~~~~  
2.2 ~Images~monochromes~~~~  
2.3 ~Images~multicouches~~~  
2.4 ~Sélection~d'une~image~  
2.5 ~Effacer~une~image~~~~~  
2.6 ~Informations~~~~~
```

--

1.32 Buffers d'images

```
2.1 Les buffers  
=====
```

Les images sont stockées sous forme de buffer: un buffer contient les données de l'image pour chaque pixel ainsi que les informations relatives à cette image (codage, niveaux de seuils, taille...).

Les buffers sont gérés à travers une liste doublement chaînée. De cette façon, chaque buffer contient deux pointeurs (sur le buffer précédent et le buffer suivant).

Lynx ne connaît donc à un moment qu'un seul buffer et va se référer à celui-ci pour parcourir toute la liste. Quand un buffer est créé il est ajouté en queue de liste.

Cette façon de faire permet d'avoir présentement en mémoire un nombre d'image important (qui ne dépend que de la capacité mémoire de la machine).

--

1.33 Images monochromes

2.2 Images monochromes ou images monocouches

=====

Les images monochromes, appelées aussi images en niveaux de gris sont donc codées dans un seul buffer et uniquement en niveaux de lumière, de 0 à 255 dans la version 8 bits, et de -32768 à +32767 dans la version 16bits.

Dans cette dernière version il peut sembler absurde d'avoir des valeurs de lumière négative, c'est pourquoi les images seront préférentiellement codées avec des valeur positives. Cependant la notion de valeurs de lumière négative prend tout son sens lorsque l'on calcul le gradient d'une image puisqu'alors l'intensité lumineuse reflète l'intensité du gradient et son signe représente le sens du gradient.

Il faut remarquer qu'aucune palette de couleur n'est attachée à une image car l'affichage (et la palette) dépend des seuils bas et haut fixés et modifiables à volonté par l'utilisateur.

--

1.34 Piles d'images

2.3 Piles d'images ou images multicouches

=====

Les piles d'images sont des empilements de couches (chaque couche étant une image monochrome). On peut les appeler également images multicouches.

Une image couleur (RVB par exemple) sont des images à 3 couches ou des piles de 3 couches.

Les couches d'une image, ou plutôt d'une pile, peuvent en outre être traitées comme des images indépendantes.

Voir aussi ~Modes~de~traitement~.

--

1.35 Sélection d'une image

2.4 Sélection d'une image

=====

Cliquez sur le bouton 'Sélection'. un requesteur apparait. Il contient une liste de toutes les images contenues en mémoire.

o En mode 'Monocouche' la liste contiendra tous les canaux de toutes les

images, car dans ce mode la notion de polychromatisme n'existe pas.

o En mode 'Pile Images' la liste piles d'images est proposée.

Il suffit de choisir l'image sur laquelle on désire travailler en la choisissant dans la liste puis 'Ok' ou en double-clickant.

--

1.36 Effacer une image

2.5 Effacer une image =====

Clicker sur le gadget représentant un visage contrarié propose l'effacement de la pile ou de la couche courante.

- o En mode 'MonoCouche', seul la couche courante sera effacée. Si, par exemple, on efface l'un des canaux d'une image trichromatique, il restera alors une image bichromatique.
- o En mode 'PileImages', toutes les couches d'une image seront éliminées de la mémoire.

--

1.37 Informations

2.6 Informations =====

Clicker sur le gadget 'I'

Une boîte de dialogue apparaît: on y trouve plusieurs champs d'informations dont les contenus ne sont relatifs qu'au buffer actuellement sélectionné.

Il peut être utile de compléter ces informations pour conserver mémoire de l'origine d'une image.

- o En mode monochrome, le nom du canal actif apparaît dans le champ 'nom'.
- o En mode 'PileImages', le nom de l'image active apparaît dans le champ 'nom'.

Le dernier champ 'historique' est complété automatiquement par le logiciel chaque fois que l'on exécute un traitement sur l'image, ceci permet donc de garder la trace des différents traitements subis par le buffer ou l'image.

Les champs d'informations sont sauvegardés sous le format 'image' uniquement.

--

1.38 Modes

Chapitre 3: Modes de traitement
=====

Note préliminaire

Tout ce qui suit ne concerne pas les spectres d'images (c'est-à-dire les images ayant subi la transformation de Fourier) et les fonctions du menu TRAITEMENTS FREQUENTIELS, qui ne fonctionnent exclusivement qu'en mode monocouche.

La fenêtre menu 'Mode' permet de fixer les condition de traitement des images mais aussi de chargement et de sauvegarde.

Mode Source

Deux positions possibles:

- > MonoCouche
- > PileImage

o MonoCouche

~~~~~

Dans ce mode toutes les couches seront traitées comme des images à part entière, donc le traitement sera appliqué aux couches individuellement.

#### o PileImages

~~~~~

Les traitements seront alors appliqués à toutes les couches de l'image simultanément.

Mode Destination

IMPORTANT: Le mode destination est ignoré si le mode source est fixé sur 'PileImages'. Un traitement sur une pile ne peut en effet donner une seule couche (sauf dans le cas d'images monochromes théoriquement). Tout se passera donc comme si le mode destination était fixé sur 'NouvPile'.

Deux positions possibles:

- > NouvCouche
- > NouvPile

o NouvCouche

~~~~~

Le traitement d'une couche créera une nouvelle couche appartenant à la même pile que la couche à partir de laquelle le traitement est effectué.

Le chargement de la prochaine image monochrome placera celle-ci sur la pile courante avec redimensionnement si nécessaire.

De même la sauvegarde au format ILF ne s'effectuera que pour la couche courante.

Cela n'a aucun effet sur la sauvegarde au format IFF.

o NouvPile

~~~~~

Le traitement créera alors une nouvelle pile à partir de la couche (ou de la pile) traitée.

Exemples

~~~~~

Soit l'image couleur toto, ses trois couches sont toto.rouge, toto.vert et toto.bleu.

- 1) Si l'on désire effectuer une convolution sur la couche toto.bleu seule,  
     créer ainsi une quatrième couche...  
     Source       = 'MonoCouche'  
     Destination = 'NouvCouche'
- 2) On désire réaliser un filtrage non-linéaire (filtre médian par exemple) sur l'image toto...  
     Source       = 'PileImages'  
     Destination = indifférent
- 3) On désire charger une image monochrome que l'on veut ajouter aux couches déjà existantes de l'image toto...  
     Source       = indifférent  
     Destination = 'NouvCouche'

Mode Masque ON/OFF

OFF : Tous les traitements sont appliqués sur toute l'image

ON : Les traitements suivants sont appliqués sur la partie de l'image sélectionnée dans l'écran de visualisation:  
     Convolution, Filtrage, non-linéaire, Addition,  
     Soustraction, Division, Ampli/Attén, Offset,  
     Transcodage.

--

## 1.39 Affichage

Chapitre 4: Affichage

=====

- 4.1 ~Choix~du~mode~d'~écran~~~~~
- 4.2 ~Réglage~des~seuils~~~~~
- 4.3 ~Transcodage~~~~~

4.4 ~Echelle~de~visualisation~~~~~  
4.5 ~Tramage~~~~~

--

## 1.40 Choix du mode d'écran

4.1      Choix du mode d'écran  
=====

Avant de visualiser une image il faut au préalable choisir le mode d'écran désiré (le mode par défaut est le mode d'écran du Workbench).

Les modes proposés sont tous des modes pour lesquels les pixels sont carrés.

On peut également choisir le nombre de couleurs affichables par l'écran, ce nombre est au minimum de 16 et au maximum de 32 (ECS/OLD) ou 256 (AGA).

De même on détermine le mode d'affichage des images en VRAIES couleurs: soit en mode HAM (6 ou 8 suivant la configuration matérielle) soit en utilisant le nombre de couleurs définit pour l'affichage fausses couleurs.

Remarque 1: La nombre de couleurs est ignoré pour l'affichage d'une image en mode HAM6 ou HAM8.

Remarque 2: On ne peut ouvrir un écran HAM6 avec une machine équipée du chipset AGA.

L'option Calcul Palette valable uniquement pour le mode d'affichage Hold And Modify sur machines AGA.

Disactiver cette option permet d'afficher les images à partir d'une palette de base fixée à l'avance.

Cela est utile pour afficher rapidement une image.

--

## 1.41 Réglage des seuils

4.2      Réglage des seuils  
=====

Ce réglage n'est disponible qu'en mode ~mode~Source~ 'MonoCouche'.

Sur la fenêtre de contrôle dans le menu 'Affichage' on trouve deux gadgets: 'seuil min' et 'seuil max'.

Ils contiennent donc les valeurs des seuils entre lesquels l'image est affichée. C'est-à-dire, pour chaque pixel, si sa valeur est inférieure à 'seuil min' il sera représenté en noir, si sa valeur est comprise entre 'seuil min' et 'seuil max' il sera représenté par une couleur, si sa valeur est supérieure à 'seuil max' il sera représenté par la couleur correspondant à 'seuil max'.

Dans le cas d'une représentation en vraies couleurs, le principe est le même (ex: pour un pixel du canal bleu, on aura: noir si  $\leq$  seuil min, un niveau de bleu si  $\text{seuil min} < \text{pixel} < \text{seuilmax}$ , Bleu maximal si  $> \text{seuil max}$ .) sauf que la couleur finale est le résultat de la composition de la couleur obtenue dans chacune des trois composantes rouge, vert, bleu.

--

## 1.42 Transcodage

### 4.3 Transcodage =====

Souvent (en astronomie notamment), on a des images mariant des zones très sombres et des zones quasi-saturées si bien qu'il est difficile de pouvoir ajuster les seuils de visualisation pour apprécier l'ensemble de l'information contenue dans l'image.

On est alors amené à réaliser un transcodage, c'est-à-dire une opération qui va associer à tous les pixels d'un certain niveau de lumière un autre niveau de lumière bien défini, et ceci pour tous les niveaux de lumière compris entre les seuils de visualisation. Cette opération est réalisée au moment de la visualisation, de sorte que le contenu de l'image ne soit pas altéré.

On pourra par exemple utiliser une table qui va relever les niveaux de lumière des faibles luminosités sans saturer les fortes lumières (fonction logarithmique), dans le cas présenté précédemment.

En pressant donc sur le bouton 'Transcodage' du menu affichage, on se retrouve devant une fenêtre qui va servir à déterminer la table à appliquer à l'image pour sa représentation.

Pour cela l'utilisateur a le choix (dans une listview) entre plusieurs fonctions primaires dont le paramètre d'intensité (c'est-à-dire l'écart qui en résultera par rapport à la fonction identité) est réglable pour certaines.

De plus, il est possible de combiner trois fonctions primaire avant d'obtenir le résultat finalement appliqué à l'image.

La table de transcodage finale est la combinaison des trois tables de base, soit:

$$\text{Table}(i) = \text{Table3}(\text{Table2}(\text{Table1}(i)))$$

pour chaque niveau de lumière  $i$ ,  $i$  variant de seuil min à seuil max.

---

Dans la fenêtre se trouve donc :

- o Un cadre servant à présenter les tables. Les fonctions primaires sont représentées en bleu et la table finale en blanc (si les deux sont confondues la table est représentée en bleu).
- o Une réglette servant à ajuster l'intensité des fonctions de base quand celles-ci le permettent.
- o Une liste permettant de choisir entre différentes fonctions.
- o Un gadget permettant de choisir la fonction (ou table) de base à éditer, parmi trois possibles.
- o Les deux boutons 'Ok' et 'Abandon' pour quitter en validant ou invalidant le choix réalisé.

En visualisation trichromie, la même table est appliquée à chacune des trois composantes, chaque composante ayant cependant ses propres valeurs de seuil.

--

## 1.43 Echelle de visualisation

### 4.4 Echelle de visualisation =====

Il arrive qu'une grande image ne soit pas visible dans sa totalité. On peut alors choisir de visualiser l'image (en pseudo-couleur ou en trichromie) à une échelle réduite (1:2 ou 1:4).

Toutes les options de visualisation et d'analyse d'image restent disponibles quelque soit l'échelle de représentation.

--

## 1.44 Tramage

### 4.5 Tramage =====

Option active uniquement pour l'affichage d'images en pseudo-couleurs.

Il s'agit d'une technique simple permettant d'obtenir l'impression visuelle d'un plus grand nombre de teintes que ce qui est réellement disponible. Cela est utile si l'on désire visualiser des images avec un mode d'affichage offrant peu de couleurs.

C'est quasiment indispensable pour les modes 16 couleurs. On arrive ainsi, avec le tramage 2x2 à 53 teintes disponibles au lieu de 14 sans tramage.

La méthode employée est celle du tramage ordonné.

Ce bouton permet de choisir entre différents modes de tramage:

- o 1x1, pas de tramage.
- o 1x2, tramage sur un pavé de 2 pixels. Permet de doubler le nombre de teintes.
- o 2x2, tramage sur un pavé de 4 pixels. Permet de quadrupler le nombre de teintes.

--

## 1.45 Visualisation

Chapitre 5: Visualisation  
=====

Lynx permet la visualisation d'images et leur analyse graphique. C'est l'objet de ce chapitre.

- 5.1 ~Visualisation~d'images~monochromes~
- 5.2 ~Visualisation~d'images~en~trichromie~
- 5.3 ~Histogramme~
- 5.4 ~Menus~de~l'écran~de~visualisation~
- 5.5 ~Palette~d'outils~de~l'écran~de~visualisation~

--

## 1.46 Visualisation d'images monochromes

5.1 Visualisation d'images monochromes  
=====

Ce mode permet la visualisation et l'analyse d'images ~monocouches~ ou de couches de piles d'images (c'est la même chose, car une couche est une image monocouche !).

- 5.1.1 ~Visualisation~en~mode~'pseudo-couleurs'~
- 5.1.2 ~Visualisation~en~mode~contour~
- 5.1.3 ~Visualisation~en~mode~mosaïque~

--

## 1.47 Visualisation en mode 'pseudo-couleurs'

### 5.1.1 Visualisation en mode 'pseudo-couleurs'

=====

Cliquer sur le bouton ~Pseudo-Coul~ ouvre l'écran de visualisation dans le mode choisit par l'utilisateur et affiche dans une fenêtre l'image en respectant les seuils min et max ainsi que la table transcodage fixée par l'utilisateur.

L'image (la couche) est donc affichée en utilisant une couleur pour représenter chaque niveau de lumière. C'est ce que l'on appelle une représentation 'pseudo-couleurs' car ici les couleurs ne représentent que des intensités de lumière.

La palette par défaut (échelle de gris) peut être éditée ou remplacée.

Pour connaître les fonctions disponibles une fois l'image affichée, voir chapitres décrivant les ~Menus~de~l'écran~de~visualisation~.

--

## 1.48 Visualisation en mode contour

### 5.1.2 Visualisation en mode contour

=====

Le mode contour est très semblable au mode pseudo-couleur, la différence la plus importante est dans la représentation de l'image, au lieu d'affecter une couleur à chaque pixel suivant son niveau, on affecte les couleurs seulement aux pixels ayant un niveau bien défini ou à la frontière séparant les pixels de niveau inférieur de ceux de niveau supérieur à ce niveau. Ceci a pour effet de tracer des isolignes, ou lignes d'égales intensité. Cela peut-être utile, notamment en astronomie, pour analyser la forme des galaxies elliptiques, par exemple.

Cliquer sur le bouton Contour, un requesteur apparaît et demande que l'on entre le nombre désiré d'isolignes, c'est-à-dire le nombre de frontières à tracer. Ce nombre peut varier de 2 à 14. Ce dernier chiffre peut paraître faible mais distinguer quelque chose lorsque le nombre d'isolignes dépasse 10 n'est déjà pas très évident (le meilleur moyen pour s'en rendre compte étant bien sûr d'essayer).

L'autre différence est que l'écran dans ce mode s'ouvre toujours en 16 couleurs, quelque soit la valeur choisie par l'utilisateur.

L'ensemble des menus de l'écran de visualisation est disponible dans ce mode. Voir le chapitres décrivant les  
~~Menus~~de~~l'écran~~de~visualisation~.

--

## 1.49 Visualisation en mode mosaïque

### 5.1.3 Visualisation en mode mosaïque =====

Ce mode de visualisation permet l'affichage simultané de plusieurs image ~monocouches~ ou de plusieurs couches de ~piles~d'images~ (le nombre de couches affichable dépend de la quantité de mémoire chip disponible).

--

## 1.50 Visualisation d'images en trichromie

### 5.2 Visualisation d'images en trichromie =====

- 5.2.1 ~Mode~'normal'~~~~~
- 5.2.2 ~Mode~Hold~And~Modify~

--

## 1.51 Mode 'normal'

### Mode 'normal' =====

Le mode 'normal' (Hold And Modify non sélectionné) affiche une image en trichromie en utilisant le nombre de couleurs déterminé avec le choix du mode écran.

Si la case 'Floyd-Steinberg' a été coché dans la fenêtre de choix du mode d'écran, alors l'affichage utilisera une méthode de tramage basée sur la diffusion d'erreurs. Cette méthode offre une meilleure qualité de rendu au prix d'un affichage plus lent.

Voir aussi ~Mode~d'écran~.

--

## 1.52 Mode Hold And Modify

### Mode Hold And Modify =====

Ce mode de visualisation d'image en trichromie est celui donnant le meilleur résultat.



Le mode Hold And Modify dépend cependant de la machine utilisée:

o Si vous êtes équipé du chipset OCS ou ECS, l’affichage se fera en mode HAM6:

- entrelacé si l'écran est en haute résolution,
- non-entrelacé si l'écran est en basse résolution.

- o Si vous êtes équipé du chipset AGA, l'affichage se fera en mode HAM8.

—

### 1.53 Histogramme

### 5.3 Histogramme

Cette fonction permet l'analyse de l'histogramme d'une image. L'histogramme est la répartition des pixels suivant leur niveau de gris. On connaît ainsi le nombre de pixels pour chaque niveau de lumière, qui sera indiqué en ordonnée.

On peut connaître précisément la population des niveaux en déplaçant la souris à l'intérieur du cadre du graphique; le nombre de pixels, ainsi que le niveau de gris pointé sont affichés au-dessus du cadre.

Grâce à l'histogramme on peut déterminer les niveaux de lumière à affecter aux seuil bas et haut de visualisation.

[illegible]

--

## 1.54 <<

Navigation à travers l'histogramme  
=====

Nécessite la version 16 bits.

Plusieurs boutons permettent de naviguer à travers l'histogramme

Loupe '+' Zoom la zone du centre de la partie visible de  
l'histogramme.

Loupe '-' Zoom Inverse.

'<' Déplacement vers la gauche de la fenêtre de visualisa-  
tion par rapport à l'histogramme.

'>' Déplacement vers la droite de la fenêtre de visualisa-  
tion par rapport à l'histogramme.

'<<' Déplacement important vers la gauche de la fenêtre de  
visualisation par rapport à l'histogramme.

'>>' Déplacement important vers la droite de la fenêtre de  
visualisation par rapport à l'histogramme.

Les bornes déterminant la zone de l'histogramme visualisée sont réglables  
directement en entrant de nouvelles valeurs:

Entre 0 et 255 pour la version 8 bits (cela n'a pas d'intérêt)

Entre -32768 et 32767 pour la version 16 bits

La borne de gauche doit avoir une valeur inférieure à la borne de droite.

--

## 1.55 Couche

Histogramme et Images multi-couches  
=====

Nécessite le ~mode~source~ PileImages

La visualisation de l'histogramme ne se fait que pour une image monochrome ou une couche seule d'une pile d'image.

Le bouton Couche permet de sélectionner l'histogramme de la couche désirée parmi toutes celle de la pile d'image active.

De plus le réglage des seuils de visualisation est par défaut celui des seuils de la couche visualisée.

Le gadget Appliquer à permet d'étendre ce réglage à toutes les couches de la pile active, c'est-à-dire que les seuils minimum et maximum fixé pour la couche dont l'histogramme est visualisé seront fixé aux mêmes valeurs pour les autres couches.

--

## 1.56 Sauver

```
Sauver
=====
```

Cette option permet la sauvegarde de la zone actuellement visible de l'histogramme sous la forme d'un fichier ASCII à deux colonnes, la première étant le niveau de gris et la seconde le nombre de pixels pour ce niveau.

## 1.57 Min Max % Niveau Nombre de points

```
Réglages des seuils pour la visualisation de l'image
=====
```

Pour régler les seuils de visualisation on peut :

- o soit mettre directement des valeurs dans les gadgets min et max
- o soit le faire graphiquement en pointant avec la souris un niveau de gris sur la zone de visualisation de l'histogramme et en cliquant :
  - avec le bouton gauche pour fixer le seuil minimum
  - avec le bouton droit pour fixer le seuil maximum

D'autres gadgets donnent quelques informations :

%

Pourcentage des pixels de l'image situés entres les seuils minimum et maximum.

---

Niveau

Niveau de gris pointé par le pointeur de la souris (si celle-ci se trouve dans la zone de visualisation de l'histogramme).

Nombre de points

Nombre de point ayant le niveau de gris définit ci-dessus

## 1.58 Menus de l'écran de visualisation

### 5.4 Menus de l'écran de visualisation

=====

Ce chapitre présente les différents menus disponibles avec l'écran de visualisation.

Projet

~Sauver~Fenetre~

~Sauver~Ecran~

-----

~Composition~

-----

~Couche~suivante~

~Couche~Prédente~

Animation

» ~Démarrer~

» ~Vitesse~~

~Libère~BitMaps~

-----

~HELP~!~

-----

~Sortir~

Palette

~Editer~

~Charger~

~Sauvegarder~

Sélection

~Information~

~Définir~Masque~

~Extraire~

~Abandonner~

~Histogramme~

Synthèse

~Tout~Effacer~

~Générer~Image~

Corrections

~Pixels~

~Ligne~

~Colonne~

```

Graphiques
~Fonte...~
-----
~Défaire~
-----
~Effacer~tous~les~graphiques~
-----
~Intégrer~dans~l'image~

```

--

## 1.59 Boîte à outils

```

Palette d'outils
=====

```

Sur l'écran de visualisation figure une palette flottante d'outils permettant l'accès aux principales fonctions d'analyse des images. Cette boîte est disposée de la façon suivante:

```

+-----+
|Tools  |
+---+---+
| A | B |
+---+---+
| C | D |
+---+---+
| E | F |
+---+---+
| G | H |
+---+---+
| I | J |
+---+---+
| K | L |
+---+---+

```

Signification des outils:

~~~~~

Outils A - J : Mode d'action de la souris

A : Sélection

Ce mode permet la sélection d'une zone rectangulaire de l'image (voir le menu 'Sélection'). Clicker et maintenir le bouton gauche permet de tracer le cadre de cette zone. Relacher pour valider. En mode 'PileImages' toutes les couches seront sélectionnées dans la région choisie.

B : Profil

Mode par défaut. Clicker et maintenir le bouton gauche permet de tracer une ligne de coupe. Le relachement de bouton va ouvrir une fenêtre représentant: en abscisse la position le long de la ligne de coupe, en ordonnée le niveau du pixel à cet endroit. Deux gadgets

permettent de fixer les valeurs correspondant au bas et haut de la fenêtre.

C : Mesure d'angle.

Pour mesurer un angle sur l'image, commencer par cliquer puis tracer une ligne jusqu'au point d'ouverture de l'angle, puis relacher la souris, une nouvelle ligne est tracé depuis le point d'ouverture jusqu'au pointeur de la souris; cliquer enfin une dernière fois donnera les informations suivantes: Angle entre les deux droites Coordonnées du point d'ouverture. Ces informations seront conservées puis proposées lors de la sélection de la ~Rotation~ d'images.

D : Synthèse

Ce mode est utilisé pour effectuer la synthèse d'images à variations lentes à partir de l'image représentée (voir menu ~Générer~Image~). Cliquer sur le bouton gauche pose une croix sous le pointeur de souris. Cliquer en maintenant la touche SHIFT enfoncée près d'une croix déjà existante permet d'oter cette dernière.

Il est conseillé de poser suffisamment de croix pour couvrir toute l'image sans pour autant que les croix se touchent entre-elles. Poser une croix tous les 16 pixels environ là où cela est possible. Le nombre maximal de croix est fixé à 256.

E : Texte (annotation)

Dans l'image, cliquer sur le bouton gauche, un requesteur apparaît pour demander l'entrée du texte à tracer.

Appuyer sur 'Ok' ou taper sur la touche 'Return' pour continuer, ou 'Cancel' pour abandonner.

Si vous validez l'entrée alors le texte apparaît sous le pointeur de souris dans la fonte de caractère choisie (voir plus loin); il suffit de cliquer à nouveau pour placer le texte à l'endroit désiré dans l'image. Il faut cependant veiller à ce que le texte ne déborde pas sur le bord droit ou vers le bas de l'image.

F : Mesure de longueur

Pour mesurer une longueur, cliquer sur la première extrémité du segment à mesurer et maintenir le bouton pressé jusqu'à la seconde extrémité. Les informations suivantes sont ensuite affichées:

Longueur du segment (en pixels)

Distance selon l'axe X des 2 extrémités

Distance selon l'axe Y des 2 extrémités

Ces deux dernières informations seront ensuite proposées lors de la sélection de la ~Translation~ d'une image.

G : Cercle (annotation)

--+

| Pour placer un élément graphique
| d'annotation sur l'image, presser
| le bouton gauche de la souris et le

H : Ligne (annotation)

| maintenir jusqu'à obtenir le dessin
| désiré.

```

I : Rectangle (annotation) | Pour plus d'informations sur les
                             | graphismes d'annotation, voir
                             | ~Graphismes~d'annotation~.
J : Ellipse (annotation)   |
                             --+

```

Outils K - L : Fenêtres supplémentaires

K : Zoom

Ouvre/Ferme la fenêtre de zoom. Pour activer le zoom maintenir enfoncé la touche 'alt gauche' et promener la souris sur l'image. On trouvera dans la fenêtre zoom la zone d'image située sous le pointeur agrandie d'un facteur 2 dans les deux directions. Pour changer le facteur d'agrandissement amener le pointeur de souris au dessus de la fenêtre de zoom et presser les touches '+' et '-'. La touche '*' permet de switcher la croix apparaissant au centre de la fenêtre de zoom.

L : Palette de couleurs

Ouvre/Ferme La fenêtre de représentation de la palette échelle de luminosité.

--

1.60 Transformations géométriques

Chapitre 6 Transformations géométriques

=====

```

6.1 ~Translation~~~~~
6.2 ~Rotation~~~~~
6.3 ~Changement~d'échelle~
6.4 ~Dimensions~~~~~

```

--

1.61 Translation

```

6.1 Translation
=====

```

Permet la translation de l'image.

Il suffit pour cela d'entrer les valeurs (en pixels) de la translation désirée. Ces valeurs peuvent être fractionnaires.

Une fois déterminées les valeurs de translation il faut choisir la méthode d'interpolation que l'on veut appliquer lorsque l'on translate de quelques fractions de pixels (0.5, 2.568...). Pour les détails voir le ~Mécanisme~de~rééchantillonnage~.

La méthode d'Interpolation bilinéaire est plus précise mais plus lente que la méthode 'Plus proche voisin'.

Note: Les paramètres de translation peuvent être pré-déterminés en visualisant l'image, voir pour cela la ~Boîte~à~outils~ (outil F) de l'écran de visualisation.

--

1.62 Rotation

6.2 Rotation =====

Permet la rotation d'une image. l'angle de rotation sera à exprimer en degrés et le centre de rotation en pixels par rapport au coin supérieur gauche de l'image. Notez que le centre de rotation peut se situer en dehors de l'image.

Une fois déterminés les paramètres de rotation, il faut choisir la méthode d'interpolation que l'on veut appliquer. Pour les détails voir le ~Mécanisme~de~rééchantillonnage~.

La méthode d'Interpolation bilinéaire est plus précise mais plus lente que la méthode 'Plus proche voisin'.

Note: Les paramètres de rotation peuvent être pré-déterminés en visualisant l'image, voir pour cela la ~Boîte~à~outils~ (outil C) de l'écran de visualisation.

--

1.63 Changement d'échelle

6.3 Changement d'échelle =====

Permet le changement d'échelle d'une image. Le changement d'échelle peut être différent selon l'axe X ou l'axe Y.

Une fois déterminés les paramètres de la transformation il faut choisir la méthode d'interpolation que l'on veut appliquer. Pour les détails voir le ~Mécanisme~de~rééchantillonnage~.

La méthode d'Interpolation bilinéaire est plus précise mais plus lente que la méthode 'Plus proche voisin'.

Note importante: ~~~~~~ Si vous effectuez un changement d'échelle en ~mode~Source~ 'MonoCouche' (donc sur une seule couche d'image),

assurez-vous d'être en ~mode~Destination~ 'NouvPile'. pour autoriser un changement de taille de l'image, dans le cas contraire, la couche image sera automatiquement redimensionnée, voir ~Redimensionnement~.

--

1.64 Redimensionnement

6.4 Dimensions =====

En mode destination 'NouvPile' uniquement, permet de changer la taille d'une image sans changement d'échelle de celle-ci.

Ceci est très utile si l'on désire, par exemple, forcer une image à être carrée pour réaliser des traitements dans le domaine fréquentiel.

--

1.65 Rééchantillonnage

Mécanisme de rééchantillonnage =====

Les transformations géométriques suivantes: Translation, Rotation, Changement d'échelle, s'effectuent suivant le modèle exposé ici.

Pour effectuer ces transformations deux options possibles:

- 1) On applique à chaque pixel de l'image de départ la transformation qui donne ses coordonnées dans l'image d'arrivée, soit:

$$\text{Nouvelle Image} = f(\text{Image originale})$$

- 2) On applique à chaque pixel de l'image d'arrivée la transformation INVERSE qui donne ses coordonnées dans l'image de départ, il suffit donc de lire la valeur du pixel dans l'image de départ pour la recopier dans l'image d'arrivée, soit:

$$\text{Image originale} = f^{-1}(\text{Nouvelle Image})$$

La méthode 1) présente un inconvénient important : il peut rester des trous dans la nouvelle image.

La méthode 2) assure que chaque pixel de la nouvelle image a trouvé son origine dans l'image de départ.

C'est donc cette dernière méthode qui est appliquée ici.

Il reste cependant un problème à résoudre: En appliquant à un pixel donné de

la nouvelle image la transformation inverse, les coordonnées correspondantes dans l'image de départ ne sont pas forcément entières, on tombe alors entre 4 pixels de l'image de départ. Quelle valeur choisir alors pour la nouvelle image ?

C'est ici qu'intervient, entre 2 possibilités, le choix de l'utilisateur :

a- Le 'Plus proche voisin'

On prend le pixel le plus proche des coordonnées déterminées par la transformation inverse. C'est la méthode la plus simple, mais pas très précise, par contre très rapide.

b- 'Interpolation bilinéaire'

On prend en compte les 4 pixels voisins des coordonnées déterminées par la transformation inverse, et l'on effectue une moyenne pondérée de leurs valeurs (les pondérations dépendant de leur distance aux coordonnées données par la transformation).

--

1.66 Traitements d'images

Chapitre 7: Traitements d'images

=====

7.1 ~Traitements~dans~le~domaine~spatial~~~~~

7.2 ~Traitements~dans~le~domaine~fréquentiel~

--

1.67 dans le domaine spatial

7.1 Traitements d'images dans le domaine spatial

=====

7.1.1 ~Convolution~discrète~~~~~

7.1.2 ~Filtrage~non~linéaire~~~~~

7.1.3 ~Transcodage~d'image~~~~~~

7.1.4 ~Opérations~arithmétiques~

7.1.5 ~Normalisation~~~~~~

--

1.68 Convolution discrète

7.1.1 Convolution discrète ou Convolution dans le domaine spatial

=====

Il y a deux types de convolutions discrètes:

```
~Convolution~par~une~matrice~
~Convolution~croisée~~~~~
```

On distingue plusieurs types de filtre pour la convolution discrète:

Les filtres 'passe-bas'. La réponse impulsionnelle d'un tel filtre est une tache plus ou moins étalée. Appliqué sur une image, il aura tendance à la rendre plus floue.

Le nom de 'passe-bas' vient du fait que ce type de filtre élimine les plus fins détails de l'image, ce que l'on assimile à des 'hautes fréquences spatiales', par analogie avec les ondes électromagnétiques où les hautes fréquences désignent des signaux à oscillations rapides. De la même manière, sur une image, les détails fins sont ceux qui évoluent le plus d'une région à une autre de l'image, donc le plus vite spatialement. Les filtres 'passe-bas' ne laissent donc passer que les détails grossiers d'une image, c'est-à-dire les formes qui évoluent plus lentement (spatialement). Bien ajusté, un tel filtre peut servir par exemple à éliminer le bruit présent dans l'image (et qui se compose essentiellement de hautes fréquences spatiales). Lynx propose tout un ensemble de filtres passe-bas.

Les filtres 'passe-haut': Ces filtres ont pour caractéristique de ne laisser passer cette fois que les hautes fréquences, c'est-à-dire dans le cas d'une image, qu'il élimine les variations lentes (détails grossiers) et ne conserve que les plus fins détails et, hélas, le bruit. Lynx propose un ensemble de filtres de caractéristiques proches du filtre passe-haut.

Les filtres 'passe-bande': Ces filtres ont pour caractéristique d'éliminer Les basses fréquences comme les filtres passe-haut, mais aussi les hautes fréquences, à la manière des filtres passe-bas. Mais il est très difficile de coder informatiquement un tel filtre, sinon en faisant appel à la transformée de Fourier, aussi n'y en-a-t-il aucun proposé dans la partie qui est consacrée au traitement d'images dans le domaine spatial.

Voir aussi ~Convolution~.

--

1.69 Convolution par une matrice

```
Convolution par une matrice
=====
```

```
La théorie
~~~~~
```

La méthode de convolution par matrice est la méthode de convolution la plus simple à mettre en oeuvre. A chaque filtre on décrit le contenu d'une matrice représentant, à peu de chose près, sa réponse impulsionnelle. Ces matrices ont une taille qui peut-être 3x3 ou 5x5

Un filtre peut être de la forme suivante (en 3x3) :

```
1 2 1
2 4 2
1 2 1
```

Le filtrage se passera de la façon suivante :

A chaque point de l'image initiale, on surperpose le filtre, et on effectue la somme pondérée de chaque pixel en correspondance avec un coefficient du filtre.

Ainsi, dans notre exemple le pixel central se verra ainsi affecter de la pondération 4, les 4 plus proches voisins seront affectés de la pondération 2 et les voisins restants de la pondération 1. On divise ensuite cette somme par la somme des coefficients pour rester dans la même dynamique que l'image initiale.

On devine donc immédiatement que ce filtre va 'étalement' le pixel central en le mélangeant avec ses voisins. Il s'agit ici d'un filtre 'passe-bas'.

Le nouveau pixel ($np[i,j]$) se verra affecté de la valeur suivante (fonction des pixels de l'image de départ, $p[i,j]$) :

```
np[i,j] = ( p[i-1,j-1]*1 + p[i,j-1]*2 + p[i+1,j-1]*1 +
            p[i-1,j] *2 + p[i,j] *4 + p[i+1,j] *2 +
            p[i-1,j+1]*1 + p[i,j+1]*2 + p[i+1,j+1]*1
            )
            / (1+2+1+2+4+2+1+2+1)
```

Dans le cas d'un filtre 5x5, les positions vont de $[i-2,j-2]$ à $[i+2,j+2]$.

Les coefficients de pondération peuvent être aussi négatifs (filtres Laplaciens, gradients) aussi peut-on ajouter une constante à l'image finale pour ne pas se retrouver avec des pixels de valeurs négatives. On remarque également que l'on va avoir un problème sur les bords de l'image. Pour s'en affranchir on n'effectue pas de traitement complet sur les 2 lignes et les deux colonnes de chaque bord. Pour une image 100x100, le traitement s'effectuera de la position 2 à 98 en x et y. Pour les bords de l'image, on remplacera les valeurs manquantes par des zéros.

On peut de plus définir un seuil bas et un seuil haut au filtrage. Ainsi les pixels en dehors de l'intervalle défini par ces deux seuils ne seront pas traités et garderont leur valeur initiale dans l'image finale.

L'option de convolution offre de nombreux filtres prédéfinis. La meilleure façon de connaître leurs effets est encore de les essayer !

La pratique
~~~~~

Pour choisir un filtre, cliquer dans la liste des filtres disponibles. Les coefficients de ce filtre s'afficheront dans les cases de la zone d'édition.

Quand un filtre est sélectionné il suffit pour le modifier de changer les valeurs inscrites dans la zone d'édition.

Signification des gadgets:

Diviseur

Constante par laquelle est divisé le produit des coefficients du filtre par les pixels de l'image. Il faut que ce nombre soit au moins égal à la somme des coefficients du filtre. Si la somme des coefficients est égal à zéro, mettre au moins '1'.

Offset

Constante qui sera ajoutée à chaque pixel de l'image finale. Cela sera utile si l'on désire ne pas se retrouver avec des valeurs négatives dans le cas des filtres 'Gradient' ou 'Laplacien'.

Seuil Bas

Valeur minimum que doit avoir un pixel pour être filtré. Si le pixel n'atteint pas ce seuil, sa valeur est simplement copiée dans l'image finale.

Seuil Haut

Valeur maximum que doit avoir un pixel pour être filtré. Si le pixel dépasse ce seuil, sa valeur est simplement copiée dans l'image finale.

Créer

Ajoute le filtre édité à la liste des filtres disponibles.

Sauver

Ecrit les filtres dans le fichier de configuration 'Lynx:Filters'.

Effacer

Effacer le filtre sélectionné.

Restore

Rétablit le filtre sélectionné à son état initial.

Si l'on clique sur 'OK', alors le filtre appliqué à l'image sera celui représenté dans la zone d'édition avec les valeurs de 'diviseur' et 'offset' inscrites dans les gadgets correspondants.

--

## 1.70 Convolution croisée

---

## Convolution croisée

=====

Dans le cas où l'on a de très grandes matrices de convolution, l'algorithme de convolution par matrice devient horriblement lent. Une méthode consiste alors à fabriquer deux filtres à une dimension, on passe alors le filtre horizontal sur l'image puis le second filtre, vertical, sur l'image filtrée par le premier filtre.

Il faut bien sûr que la fonction mathématique correspondant au filtre soit séparable, c'est-à-dire que le traitement puisse ce faire indifféremment selon les lignes puis selon les colonnes.

C'est le cas de la fonction gaussienne

Ce filtre particulier est un filtre passe-bas, c'est-à-dire qui ne laisse passer que les basses fréquences de l'image, donc ses variations lentes. Les détails fins (hautes fréquences) sont ainsi éliminés. Le résultat est une image lissée.

L'intérêt du filtre gaussien vient du fait qu'il est possible de lisser très fortement l'image, bien plus qu'en utilisant le filtrage par matrice de convolution, et ce pour un temps de calcul équivalent.

On demande l'entrée d'un paramètre, sigma, qui correspond à la largeur de la gaussienne que l'on va passer sur l'image. Plus sigma est grand plus l'image sera lissée. sigma peut être compris entre 0 et 5.

Cette option est surtout utilisée pour réaliser un filtrage par masque flou, dont le but est de renforcer les contrastes dans l'image. Le principe en est le suivant:

On effectue la différence entre l'image à traiter et cette même image passée à travers un filtre gaussien, le résultat est une image ne contenant que les hautes fréquences (les détails fins) de l'image initiale. On amplifie en suite ce résultat, par exemple au moyen d'une ~Multiplication par une constante~. Enfin on additionne cette dernière image à l'image initiale et l'on obtient une image où les hautes fréquences ont été fortement accentuées.

--

## 1.71 Filtrage non linéaire

### 7.1.2 Filtrage non linéaire

=====

Les filtres non linéaires sont assez semblables aux matrices de convolution sauf que leur effet n'est plus linéaire, en ce sens que l'effet du filtre dépend des valeurs des pixels et de leur environnement.

Les filtres non linéaires utilisés ici sont constitués par une matrice 5x5

---

qui définit un 'élément structurant':

le filtre peut avoir la forme suivante:

```
00000
00X00
0XXX0
00X00
00000
```

La forme de l'élément structurant est définie par les coefficients de la matrice qui sont marqués à un X. Le principe de fonctionnement est le suivant: A chaque pixel de l'image, on superpose le filtre. Chaque point situé sous la matrice est pris en compte suivant que le coefficient de la matrice est ou non égal à 'X'. Et l'on affecte alors au pixel central une valeur dépendant de lui-même et des ses voisins.

La forme de l'élément structurant est laissée au libre choix de l'utilisateur. Lynx propose par défaut une forme simple et souvent utilisé.

Trois 'effets' sont prédéfinis:

Le filtre 'Minimum', qui affecte au pixel central la plus petite valeurs des pixels pris en compte. Ce filtre a pour effet d'éroder l'image.

Le filtre 'Maximum', qui affecte la plus grande valeur. Ce filtre a pour effet de dilater l'image.

Le filtre 'Median', qui retourne la valeur médiane. Ce filtre demande que l'on entre un coefficient d'efficacité, qui varie de 0 (effet presque nul) à 1 (effet maximal). Le filtre median est tres efficace pour atténuer le bruit et éliminer des pixels aberrants.

Il est possible de passer ces filtres plusieurs fois de suite sur une image; il faudra cependant modifier la forme de l'élément structurant car les contours de l'image ont tendance à prendre une forme qui y ressemble.

On peut effectuer des combinaisons de filtres:

Une 'Ouverture', qui résulte du passage successif d'un filtre maximum puis du filtre minimum.

Une 'Fermeture' qui résulte du passage successif du filtre minimum puis du filtre maximum.

--

## 1.72 Transcodage d'image

### 7.1.3 Transcodage =====

Le transcodage est un moyen d'agir directement sur l'histogramme d'une

---

image, en associant à chaque valeur de pixel une nouvelle valeur préétablie.

Ici, contrairement au transcodage du menu de visualisation, on crée une nouvelle image, qui est la transcodée de l'image d'entrée, soit on remplace l'image d'entrée par sa transcodée.

L'image est donc réellement affectée par ce traitement.

Lynx permet de coder les images sur 256 (vers. 8bits) ou 65536 (vers. 16bits) niveaux de gris, la table de transcodage sera donc un tableau de 256 ou 65536 valeurs, soit:

$$V_i = \text{LUT}[i]$$

Où  $i$  représente la valeur du pixel à transcoder et  $V_i$  sa nouvelle valeur issue de la table de transcodage  $\text{LUT}[]$ .

#### ATTENTION

Si l'image en question est un spectre en représentation polaire (module et argument) alors le transcodage affectera également le MODULE de ce spectre suivant le résultat de l'opération sur la zone image, soit

$$\text{Nouvelle\_valeur} = \text{Ancienne\_valeur} * \frac{\text{Nouvelle\_valeur\_pixel\_image}}{\text{Ancienne\_valeur\_pixel\_image}}$$

Il existe donc plusieurs fonctions de transcodage:

#### Linéaire

Rééchantillonne entre les valeurs de sorties minimum et maximum la partie de l'histogramme située entre les valeurs d'entrée minimum et maximum. On peut ainsi réaliser un négatif, par exemple, en fixant un seuil de sortie max inférieur au seuil de sortie min.

#### Log

Réhausser les faibles valeurs sans saturer les fortes valeurs. Elle permet de mieux adapter l'image à l'oeil.

#### Egalisation d'histogramme

Réalise une nouvelle répartition de l'histogramme en cherchant à l'uniformiser au mieux (de telle sorte, d'un point de vue statistique, que chaque pixel ait la même probabilité d'avoir tel ou tel niveau de gris).

#### Seuillage bas

Réalise un seuillage de l'image en remplaçant tous les pixels en dessous de la valeur d'Entrée minimum par la nouvelle valeur Seuil.

#### Seuillage haut

Réalise un seuillage de l'image en remplaçant tous les pixels au dessus de la valeur d'Entrée maximum par la nouvelle valeur Seuil.



#### Troncature

Coupe l'histogramme pour ne garder que les valeurs situées entre les valeurs d'entrée minimum et maximum. Les valeurs situées au-delà de l'une des deux bornes sont ramenées à la valeur de la borne la plus proche.

#### ET Logique

Réalise l'opération ET logique entre chaque valeur de gris et la valeur entrée dans Coefficient, cela permet de discrétiser fortement l'histogramme.

#### Rotation

Effectue une rotation de l'histogramme dans le domaine des valeurs POSITIVES uniquement. La valeur de 'coefficient' détermine l'amplitude (en niveau de gris) positive ou négative de la rotation.

--

## 1.73 Opérations arithmétiques

### 7.1.4 Opérations arithmétiques

=====

~MultConstant~ Multiplication par une constante  
 ~AddConstante~ Ajout d'une constante  
 ~Addition~~~~~ Addition d'une image  
 ~Soustraction~ Soustraction d'une image  
 ~Division~~~~~ Division d'une image

--

## 1.74 MultConstant

Multiplication par une constante  
 =====

Multiplie les pixels de l'image active par une constante réelle.

Etant donnée que chaque pixel est affecté indépendamment des autres, il est possible de ne pas créer une nouvelle image, cliquez alors sur 'Même Image', sinon sur 'Nouvelle Image'.

--

## 1.75 AddConstante

Addition d'une constante  
=====

Ajoute une constante au pixels de l'image.

Etant donnée que chaque pixel est affecté indépendamment des autres, il est possible de ne pas créer une nouvelle image, cliquez alors sur 'Même Image', sinon sur 'Nouvelle Image'.

--

## 1.76 Addition

Addition avec une autre image  
=====

Permet l'addition de l'image courante avec une autre image.

La première condition est que les tailles soient identiques.

Si le ~mode~Source~ est 'PileImages' il est alors nécessaire que les deux images aient le même nombre de couches, autrement l'addition est impossible.

Il faut enfin choisir le mode d'addition désiré:

- o Simple                    > addition ordinaire de chaque pixel
- o Moyenne                 > Moyenne des pixels des deux images
- o Module                  > racine carrée de la somme des carrés pour chaque pixel  
ATTENTION: En version 8bits les valeurs des pixels sont  
considérées comme étant signées soit : 0 - 255 correspondra  
à (-128) - 127 il y aura donc un décalage de 128

--

## 1.77 Soustraction

Soustraction  
=====

Permet la soustraction d'une image de l'image active.

La première condition est que les tailles soient identiques.

Si le ~mode~Source~ est 'PileImages' il est alors nécessaire que les deux images aient le même nombre de couches, autrement la soustraction est

impossible.

--

## 1.78 Division

Division  
=====

Permet la division de l'image active par une autre image

La première condition est que les tailles soient identiques.

Si le `~mode~Source~` est 'PileImages' il est alors nécessaire que les deux images aient le même nombre de couches, autrement la division est impossible.

--

## 1.79 Normalisation

7.1.5 Normalisation  
=====

Cette opération va redistribuer l'échelle de gris de (ou des) la couche(s), suivant l'état du `~mode~Source~`, entre les valeurs minimale et maximale déterminées par l'utilisateur.

--

## 1.80 dans le domaine fréquentiel

7.2 Traitement d'images dans le domaine fréquentiel  
=====

7.2.1 ~Transformation~de~Fourier~  
7.2.2 ~Convolution~  
7.2.3 ~Déconvolution~  
7.2.4 ~Corrélation~  
7.2.5 ~Convolution~par~une~fonction~analytique~

--

## 1.81 Transformation de Fourier

### 7.2.1 Transformation de Fourier

=====

~Transformée~dans~le~sens~directe~

~Transformée~dans~le~sens~inverse~

--

## 1.82 Transformée directe

Transformée directe

=====

Ne fonctionne qu'en ~mode~Source MonoCouche.

Cette fonction calcule la transformée de Fourier de l'image active. L'image devra toutefois remplir la condition suivante, à savoir que ses dimensions doivent OBLIGATOIREMENT être des puissances de 2 identiques (128x128, 256x256, 512x512...).

La transformée de Fourier permet de passer de la représentation spatiale d'une image à sa représentation fréquentielle... On obtient par cette opération une image représentant des amplitudes dans le domaine des fréquences spatiales: les basses fréquences (variations lentes dans l'image) seront représentées au centre de l'image et les fréquences seront d'autant plus grandes que l'on s'écarte du centre de l'image.

En fait Lynx va allouer l'espace nécessaire pour y placer la TF indépendamment de la mémoire déjà allouée pour l'image.

La mémoire à allouer par la TF est très importante car il faut stocker la partie réelle et la partie imaginaire de la TF sous forme de nombre flottant : soit en tout 8 octets en plus par pixel.

Pour permettre la visualisation de la transformée de Fourier le logarithme du module de la TF est copiée dans la zone image (qui est la zone normale de stockage de l'image). On prend le logarithme car la dynamique d'un spectre est très forte, cela permet une visualisation plus aisée.

--

## 1.83 Transformée inverse

Transformée inverse

=====

Effectue la transformation du domaine spectral vers le domaine spatial de

l'image.

Deux questions seront posées à l'utilisateur au cours de cette opération:

o Doit-on faire une normalisation ?

Si oui Lynx va étaler l'échelle de gris de l'image entre les niveaux minimum et maximum (0 à 255 en 8bits ou -32768 à +32767 en 16bits).

o Doit-on faire une permutation ?

Si l'on a effectué une convolution ou une corrélation avec un autre spectre alors il faut répondre oui.

Après la transformation, les zones allouées pour le stockage du spectre sont désallouées.

--

## 1.84 Théorie

Convolution  
=====

La convolution est une opération mathématique qui réalise un filtrage sur une fonction d'entrée définie.

Dans de nombreuses mesures physiques intervient le phénomène suivant: on introduit à l'entrée d'un appareil la grandeur 'e' à mesurer et à la sortie on recueille une fonction 's' de la forme:

$$s(t) = \int_{-\infty}^{+\infty} h(t-u) e(u) du = \int_{-\infty}^{+\infty} h(u) e(t-u) du$$

où s, e et h sont des grandeurs dépendantes du temps t (par exemple).

Symboliquement:

$$e(t) \rightarrow \text{---} \int \text{---} h(t) \text{---} \rightarrow \text{---} s(t)$$

où h représente la fonction caractéristique de l'appareil. Cette opération s'appelle la convolution par h.

On note :  $s(t) = e(t) * h(t)$

où '\*' est l'opérateur de convolution.

Dans le domaine des fréquence la convolution est une simple multiplication des deux spectres, soit :

$$S(f) = E(f) \cdot H(f)$$

où  $S(f)$ ,  $E(f)$  et  $H(f)$  sont les transformées de Fourier respectives de  $s(t)$ ,  $e(t)$  et  $h(t)$ . D'où l'intérêt de passer dans le domaine des fréquences !

Imaginons par exemple le cas où l'appareil est un appareil photographique.

On réalisera l'opération suivante :

$$(\text{photo}) = (\text{Image de l'objet à l'entrée}) * (\text{Appareil photographique})$$

La caractéristique principale du filtre est sa 'réponse impulsionnelle', c'est-à-dire le signal qui sortira du filtre si on y met en entrée une impulsion. Dans le cas d'une image on regardera ce qui se passe si en entrée du filtre cette impulsion est représentée par un seul point de forte intensité lumineuse.

Il existe deux façons de réaliser la convolution :

- 1- dans le domaine spatial direct
- 2- dans le domaine fréquentiel

1- ~Convolution~dans~le~domaine~spatial~:

On réalise la convolution d'une image par une matrice représentant la réponse impulsionnelle.

Cette façon de faire est très économique dans le cas de réponse impulsionnelles simples nécessitant des matrice de petites tailles, mais n'est pas adaptée si la réponse est plus complexe et demande de grande matrices, on a alors recours à la

2- ~Convolution~dans~le~domaine~fréquentiel~:

On peut réaliser de cette manière la convolution d'une image :

- o soit avec une fonction qui va filtrer certaines fréquences de l'image.
- o soit avec une autre (elle doivent être de même taille) dont l'une représentera la réponse impulsionnelle du filtre qui peut alors être très complexe.

--

## 1.85 Convolution

### 7.2.2 Convolution =====

~Théorie~  
~~~~~

Pratique
~~~~~

Dans l'espace fréquentiel la convolution entre deux image se réduit aux produit de leur spectre, ce qui est très rapide, d'où l'intérêt de la transformée de Fourier.

On réalise ici la convolution entre deux images ou la première (l'image active) est l'image à convoluer, et la seconde est la réponse impulsionnelle.

Il est nécessaire avant de procéder à la convolution que les deux images aient subi la transformation de Fourier dans le sens directe.

Le produit de convolution sera stockée à la place de la seconde image (la réponse impulsionnelle). Cela signifie qu'en cas de résultat insatisfaisant, vous devrez recharger l'image de la réponse impulsionnelle. Le plus sage est donc d'en conserver une copie prête pour une ré-utilisation.

--

## 1.86 Déconvolution

Déconvolution  
=====

La déconvolution est l'opération inverse de la ~convolution~.

Reprenons l'exemple vue dans le chapitre sur la convolution où l'on a un signal 's(t)' de sortie, 'e(t)' d'entrée et une fonction d'appareil 'h(t)':

```

      +-----+
e(t) -->--|  h(t)  |-->-- s(t)
      +-----+

```

Dans le cas de nos images, on aura:

```

      f(x,y) = g(x,y) * h(x,y)
      ||
image observée

```

Dans le domaine fréquentiel, après transformée de Fourier, on a:

$$F(u,v) = G(u,v) \cdot H(u,v)$$

où u et v représentent l'espace des fréquences spatiales à 2 dimensions de nos images.

Le plus souvent l'on va mesurer un signal de sortie issu du détecteur (appareil photo, caméra CCD). Ce signal a été perturbé lors de son passage par le détecteur et d'autres éléments potentiellement perturbateurs

(atmosphère terrestre par exemple). Ce sont toutes ces perturbations que l'on va retrouver dans  $h(x,y)$ .

L'objectif de la déconvolution est d'essayer de retrouver le signal (dans notre cas une image) tel qu'il aurait été obtenu en l'absence de toute perturbation.

La déconvolution peut s'écrire dans le domaine fréquentiel:

$$G(u,v) = \frac{F(u,v)}{H(u,v)}$$

Cela s'appelle filtrage inverse. Il suffit donc de connaître, en plus du signal de sortie, la fonction d'appareil représentant la perturbation pour en déduire  $E$ .

Dans la pratique ce n'est hélas pas si simple. On voit tout de suite que quand  $H(u,v)$  prend des valeurs nulles la déconvolution est impossible.

En fait on considère qu'il existe un bruit additif  $N$  présent dans les images:

$$F(u,v) = G(u,v) \cdot H(u,v) + N(u,v)$$

Le bruit  $N(u,v)$  étant difficile à quantifier on ne peut restituer l'image originale  $g(x,y)$  que de manière approchée.

La méthode employée par Lynx est le filtrage de WIENER (ou restauration par les moindres carrés). Elle consiste à écrire l'image restaurée sous la forme:

$$G(u,v) = \frac{F(u,v) \cdot H^*(u,v)}{H(u,v) \cdot H^*(u,v) + k}$$

avec  $H^*(u,v)$  le complexe conjugué de  $H(u,v)$  et ' $k$ ' un paramètre égal au rapport du spectre de puissance du bruit et du spectre de puissance du signal.

Le bruit étant difficile à quantifier on détermine  $k$  par tâtonnements de façon à atteindre le résultat le plus satisfaisant (valeur typique de départ:  $k = 0.0001$ ). Si  $k = 0$ , alors le filtrage de Wiener est identique au filtrage inverse.

Dans le cas d'images astronomiques, il suffit de choisir une étoile (non saturée) pour représenter assez correctement la fonction  $h(x,y)$ .

--

## 1.87 Corrélation



#### 7.2.4 Corrélation ou Intercorrélation

=====

La corrélation est une opération permettant de mesurer la 'ressemblance' de deux images.

La méthode la plus simple consiste à décaler les deux images l'une par rapport à l'autre et de mesurer le degré de similitude en fonction du décalage.

Dans le domaine fréquentiel, la fonction d'intercorrélation s'obtient par :

$$A(u,v) = I1(u,v) \cdot I2^*(u,v)$$

où I1 et I2 sont les deux images à comparer, I2\* le complexe conjugué de I2 et A la fonction d'intercorrélation.

La corrélation est une opération assez proche de la convolution puisque l'on réalise ici le produit de l'image par le complexe conjugué du spectre. C'est aussi la raison pour laquelle cette opération est difficilement réalisable dans le domaine spatial.

--

## 1.88 Convolution par une fonction analytique

#### 7.2.5 Convolution par une fonction analytique

=====

Cette fonction permet la ~convolution~ directe du spectre avec une fonction de filtrage des fréquences.

Lynx offre les fonctions suivantes :

- o Low-Pass  
Filtre passe-bas réalisant une coupure brutale des fréquences situées au-delà de la fréquence de coupure. Ce filtre a surtout une valeur pédagogique.
- o Low-Pass Gauss  
Filtre Passe-bas dont la forme est celle d'une courbe de Gauss. Ce filtre réalise correctement son rôle de filtrage passe-bas.
- o Low-Pass Butterworth  
Filtre Passe-bas.
- o High-Pass  
Filtrage passe-haut. Ne laisse donc passer que les hautes

fréquences.

- o Wavelet  
Filtre réalisant l'analyse en ondelettes. Ce filtre a pour vocation de ne laisser passer qu'une gamme de fréquences représentatives d'une certaine classe de structures dans l'image
- o Hanning
- o Blackman

--

## 1.89 LynxFormat

LynxFormat  
=====

Il s'agit du format standard de Lynx (si, si !).

Il supporte:

- o Les images codées sur 8 ou 16 bits par pixel
- o les images mono- ou multi-couches.

--

## 1.90 Format IFF-ILBM

Format IFF-ILBM  
=====

Lynx supporte les formats IFF suivants en lecture:

- o IFF 1-8 bits
- o IFF 24 bits

Lynx supporte les formats IFF suivants en écriture:

- o IFF 1-8 bits (Uniquement à partir de l'écran de visualisation)
- o IFF HAM6 et HAM8 (Uniquement à partir de l'écran de visualisation)
- o IFF 24 bits (Uniquement à partir de l'écran de contrôle)

--

---

## 1.91 FITS

Format FITS  
=====

FITS signifie: Flexible Image Transport System

Il s'agit du format d'image utilisé par la communauté des astronomes professionnels notamment.

On peut trouver des images au format FITS sur certains sites FTP (Nasa par ex.).

Lynx supporte le FITS pour les images monochromes codées sous forme d'entiers 8 ou 16 bits par pixel en lecture et en écriture.

--

## 1.92 IMA

Format IMA  
=====

Le format 'Ima' est un format simple pour stocker une image monochrome codée en 8 ou 16 bits signés ou non.

Un fichier IMA contient:

- o Un entête de 6 octets pour indiquer le type de l'image ( 8 ou 16 bits, signé ou non) et ses dimensions:
  - 1 UWORD : largeur
  - 1 UWORD : hauteur
  - 1 UBYTE : Type de l'image (bit 0 = 1 > signé, non signé sinon
    - bit 1 = 1 > pas de valeurs négatives
    - bit 7 = 1 > données sur 16 bits, 8 bits sinon)
  - 1 UBYTE : nombre de plans de bits (non utilisé)
- o Ensuite viennent les données de l'image (largeur\*hauteur pixels) balayée ligne par ligne de gauche à droite et de haut en bas.

--

## 1.93 PIC

Format PIC  
=====

Le format PIC est le format du logiciel MiPS (Micro image Processing System) disponible sur PC et compatibles

Ce format ne gère que des images monochromes codées sur 16 bits.

Il n'est supporté qu'en lecture par Lynx.

--

## 1.94 Objet (datatypes)

Les Datatypes  
=====

Voir Ouvrir sous...~Objet~.

Vous y trouverez la description du support des datatypes.

--

## 1.95 Utilisation d'ARexx

Chapitre 8: Utilisation d'ARexx  
=====

Le port ARexx n'est que partiellement opérationnel, les fonctions disponibles étant peu nombreuses:

```
~AddImage~~~~~
~SubImage~~~~~
~AddrCurrentImage~
~SelectImage~~~~~
~Convolve~~~~~
~CurrentImageName~
~DeleteBuffer~~~~~
~Normalize~~~~~
~RotateImage~~~~~
~Mult~~~~~
~Mask~~~~~
~MaskStatus~~~~~
~Sigma~~~~~
~LynxEnd~~~~~
```

Les scripts ARexx écrits pour Lynx sont tous batis sur le meme modèle (Template.imrx), qui assure une gestion efficace des erreurs.  
De la même façon tout nouveau script doit être construit à partir du modèle 'Template.imrx'.

Lynx renvoie un code de retour après la demande d'exécution de chaque

commande. Voir ~Messages~d'erreurs~.

--

## 1.96 AddImage

Commande ARexx  
=====

### NAME

ADDIMAGE -- Additionne une image à l'image active

### SYNOPSIS

ADDIMAGE BUFFER/A METHOD

### FUNCTION

Cette Fonction ajoute une image à l'image active selon la méthode choisie par l'utilisateur. Par défaut une addition simple est effectuée et les nouvelles valeurs dépassant la dynamique autorisée suivant la version de Lynx sont fixées à la valeur maximale.

La méthode 'AVERAGE' effecute la moyenne entre les deux images.

La méthode 'MODULE' effectue la somme en module soit :

$$S = \text{sqrt}(p1 \times p1 + p2 \times p2)$$

où p1 et p2 représentent les intensités à additionner. Dans la version 8bits de Lynx l'addition en module suppose que les valeurs inférieures à 128 soient en fait négatives, on passe donc en une représentation signée.

### INPUTS

BUFFER - Adresse de l'image à additionner.  
METHOD - Méthode d'addition.

### RESULT

### NOTE

### SEE ALSO

--

## 1.97 SubImage

---

Commande ARexx  
=====

#### NAME

SUBIMAGE -- Soustrait une image de l'image active

#### SYNOPSIS

SUBIMAGE BUFFER/A

#### FUNCTION

Cette Fonction soustrait une image de l'image active.  
Les nouvelles valeurs dépassant la dynamique autorisée par valeur inférieure sont mises au niveau minimum.  
Les nouvelles valeurs dépassant la dynamique autorisée par valeur supérieurs sont mises au niveau maximum.

#### INPUTS

BUFFER - Adresse de l'image à additionner.

#### RESULT

#### NOTE

#### SEE ALSO

--

## 1.98 AddrCurrentImage

Commande ARexx  
=====

#### NAME

ADDRCURRENTIMAGE -- Renvoie l'adresse de l'image active.

#### SYNOPSIS

ADDRCURRENTIMAGE VAR BUFFER

#### FUNCTION

Cette fonction renvoie l'adresse de l'image active.

#### INPUTS

#### RESULT

---

BUFFER - Adresse de l'image

NOTE

SEE ALSO

--

## 1.99 SelectImage

Commande ARexx  
=====

NAME

SELECTIMAGE -- Sélectionne l'image à activer

SYNOPSIS

SELECTIMAGE NAME/A

FUNCTION

Cette fonction sélectionne l'image nommé nom. Cette dernière devient alors l'image active.

INPUTS

NAME : Nom de l'image à activer.

RESULT

NOTE

SEE ALSO

~CURRENTIMAGENAME~

--

## 1.100 Convolve

Commande ARexx  
=====

NAME

CONVOLVE -- Effectue la convolution d'une image par une matrice

## SYNOPSIS

```
CONVOLVE FILTRE/A SEUIL_BAS/N SEUIL_HAUT/N
```

## FUNCTION

Cette fonction permet la convolution de l'image active dans le domaine spatial avec une matrice de convolution .

Les matrices disponibles sont les mêmes que celle disponibles avec la fonction de convolution dans le domaine spatial.

Les paramètres SEUIL\_BAS et SEUIL\_HAUT sont optionnels et indiquent que la convolution sera réalisée sur les pixels dont l'intensité est comprise entre SEUIL\_BAS et SEUIL\_HAUT.

Si les paramètres SEUIL\_BAS et SEUIL\_HAUT ne sont pas indiqués alors tous les pixels seront traités.

## INPUTS

```
FILTRE      - Nom de la matrice à employer
SEUIL_BAS   - Seuil bas sur les intensités.
SEUIL_HAUT  - Seuil haut sur les intensités.
```

## RESULT

## NOTE

## SEE ALSO

--

## 1.101 CurrentImageName

```
Commande ARexx
=====
```

## NAME

```
CURRENTIMAGENAME -- Renvoie le nom de l'image active
```

## SYNOPSIS

```
CURRENTIMAGENAME VAR NOM
```

## FUNCTION

Cette Fonction renvoie le nom de l'image active.

## INPUTS



RESULT  
NOM - Nom de l'image active

NOTE

SEE ALSO

--

## 1.102 DeleteBuffer

Commande ARexx  
=====

NAME  
DELETEBUFFER -- Efface une image de la mémoire.

SYNOPSIS  
DELETEBUFFER BUFFER/A

FUNCTION  
Cette Fonction efface une image de la mémoire de Lynx.

INPUTS  
BUFFER - Adresse de l'image à effacer.

RESULT

NOTE

SEE ALSO

--

## 1.103 Normalize

Commande ARexx  
=====

NAME  
NORMALIZE -- Normalisation de l'image active.

SYNOPSIS  
NORMALIZE MIN/A MAX/A

**FUNCTION**

Cette Fonction redistribue l'échelle des intensités lumineuses de l'image active entre les bornes MIN et MAX.

**INPUTS**

MIN - Borne supérieure de la normalisation.  
MAX - Borne inférieure de la normalisation.

**RESULT****NOTE**

Si les bornes MIN et MAX sont incompatibles ( $MIN > MAX$ ) où au delà de la dynamique autorisée, une erreur se produit et le traitement est interrompu.

**SEE ALSO**

--

## 1.104 Rotatelmage

Commande ARexx  
=====

**NAME**

ROTATE -- Rotation de l'image active

**SYNOPSIS**

ROTATE ANGLE/A XC/A YC/A METHOD/K

**FUNCTION**

Cette Fonction effectue la rotation de l'image autour du point de coordonnées spécifié par XC et YC.

Il est possible de spécifier la méthode de rééchantillonnage à employer. La méthode faisant appel à l'interpolation offre le meilleur résultats au prix d'un plus long temps de calcul.

**INPUTS**

BUFFER - Adresse de l'image à additionner.  
METHOD - Méthode de rééchantillonnage: I avec Interpolation  
N sans interpolation

**RESULTS**

## NOTE

Les coordonnées XC et YC peuvent se situer en dehors de l'image

## SEE ALSO

~Rotation~

--

## 1.105 Mult

Commande ARexx

=====

## NAME

MULTCONST -- Multiplie l'image active avec une constante

## SYNOPSIS

MULTCONST BUFFER/A CONST/A

## FUNCTION

Cette Fonction affectue le produit de l'image avec une constante réelle de valeur quelconque choisie par l'utilisateur.

## INPUTS

BUFFER - Adresse de l'image à additionner.  
CONST - La constante multiplicative

## RESULT

## SEE ALSO

--

## 1.106 Mask

Commande ARexx

=====

## NAME

MASK -- Fixe l'état du masquage.

## SYNOPSIS

MASK ONOFF/A

## FUNCTION

Permet d'activer/désactiver le masquage

#### INPUTS

ONOFF - Etat désiré : ON > activé.  
OFF > désactivé.

#### RESULT

#### NOTE

#### SEE ALSO

~Masquage~

--

## 1.107 Mask\_Status

Commande ARexx  
=====

#### NAME

MASKSTATUS -- Interroge Lynx à propos de l'état du masquage

#### SYNOPSIS

MASKSTATUS VAR Masque

#### FUNCTION

Cette Fonction renvoie dans la variable 'Masque' l'état du masquage, c'est-à-dire son état d'activation ou d'inactivation.

#### INPUTS

#### RESULT

L'état du masquage:  
ON  
ou OFF

#### NOTE

#### SEE ALSO

~Masquage~

--

---

## 1.108 Sigma

Commande ARExx  
=====

### NAME

SIGMA -- Calcul l'écart type de l'image active

### SYNOPSIS

SIGMA VAR Sigma

### FUNCTION

Cette Fonction effectue le calcul de l'écart type de l'image active et renvoie le résultat dans la variable 'Sigma'.  
Cette fonction ne doit être appelée qu'en mode Monocouche.

### INPUTS

### RESULT

La valeur de l'écart type.

### NOTE

Si l'on veut que le calcul se fasse seulement dans une zone de l'image. Il faut au préalable avoir défini un masque et l'avoir activé.

### SEE ALSO

--

## 1.109 LynxEnd

Commande ARExx  
=====

### NAME

LYNXEND -- Termine un script AREXX

### SYNOPSIS

LYNXEND

### FUNCTION

Cette commande doit OBLIGATOIREMENT être envoyé à Lynx à la fin de tout script AREXX.

### INPUTS

RESULT

NOTE

SEE ALSO

--

## 1.110 Sauver Fenêtre

Sauver Fenêtre  
=====

Permet de sauver au format IFF le contenu de la fenêtre active. Cette fenêtre n'est donc pas systématiquement celle contenant l'image visualisée mais n'importe laquelle des fenêtres présente sur l'écran.

L'image sera sauvée dans le même mode que l'écran de visualisation.

--

## 1.111 Sauver Ecran

Sauver Ecran  
=====

Permet de sauver au format IFF le contenu de l'écran de visualisation.

--

## 1.112 Composition d'images

Composition d'images  
=====

Objectif  
~~~~~

L'objectif initial de cette fonction est de pouvoir réaliser des mosaïques d'image, comme par exemple dans le cas où l'on aimerait fusionner deux images ayant une partie commune.

Cependant l'analyse de l'ensemble des cas de figure a amené à réaliser une fonction plus générale; il est donc possible de réaliser non seulement des mosaïques mais aussi certains effets spéciaux en incluant une image dans une autre et jouant sur les paramètres de transition entre les deux images.

Procédure

~~~~~

- 1/ Sélection de l'image à intégrer (2) à l'image visualisée (1).  
l'image sélectionnée est dessinée dans une fenêtre sans aucune bordure
- 2/ Placement de l'image choisie par rapport à l'image visualisée:  
pour cela déplacer la fenêtre sans bordure comme une fenêtre ordinaire,  
la barre de déplacement est invisible mais bien présente.
- 3/ S'assurer que la fenêtre de l'image (2) est active et cliquer sur le bouton de menu.
- 4/ La fenêtre de contrôle de Composition est affichée:

```

+-----+
| Composition                                     |
+-----+
| +---Méthode---+ +-----Transition-----+ |
| | () Image 1   | |                               | |
| | () Image 2   | | Image 1   Linéaire           | |
| | () Minimum   | |                               | |
| | () Maximum   | | Image 2   Linéaire           | |
| | () Mixage     | |                               | |
| | () Addition   | +-----+                     |
| +-----+                                     |
|                               Niveau Ailleurs |   |
| Abandon                               |   |
|                               Seuil Minimal  |   |
| OK                               |   |
|                               Mixage (%)    |   |
+-----+

```

- o Niveau ailleurs détermine la valeurs à affecter aux pixels en dehors des deux images.
- o Seuil Minimal détermine la valeur minimale que doivent avoir les pixels de l'image 2 pour être pris en compte pour le traitement.
- o Mixage (%) détermine le taux de mixage de l'image 1 par rapport à l'image 2 lorsque la méthode de composition choisie est 'Mixage'.

**IMPORTANT:** En mode Source 'multicouche' la composition peut s'appliquer à l'ensemble des couches de l'image (1) si l'une des deux conditions suivantes est remplie:

- o Le nombre de couche de l'image (2) est le même que celui de l'image (1), auquel cas la composition est effectuer couche à couche.
- o L'image (2) est monocouche et la composition de chaque couche de l'image (1) est effectuée avec l'unique couche de l'image (2).

--

### 1.113 Méthode

### Méthode de composition

=====

Décrit la méthode à appliquer à la zone de recouvrement des deux images à composer.

Admettons que l'on se trouve dans la configuration suivante, où (1) désigne l'image origine, et (2) l'image choisie pour la composition :

```

+-----+
|         |
|   (1)  +-----+
|         | (2)  |
|         |     |
|         +-----+
|         |     |
+-----+

```

Les différentes méthodes applicables sont les suivantes:

- o Image 1 : les pixels de l'image (1) viennent remplir cet espace.
- o Image 2 : les pixels de l'image (2) viennent remplir cet espace.
- o Minimum : la valeur minimum entre les chaque pixel de l'image (1) et de l'image (2).
- o maximum : la valeur maximum entre les chaque pixel de l'image (1) et de l'image (2).
- o Mixage : Moyenne pondérée entre les pixels de l'image (1) et de l'image (2). Cette méthode est applicable seulement dans le cas où l'image (2) est centièrement comprise à l'intérieur de l'image (1).
- o Addition: Addition des valeurs des pixels de l'image (1) et de l'image (2).

--

## 1.114 Transition

### Méthode de transition

=====

Décrit comment doit se faire la transition d'une image vers l'autre

Les différentes transition possibles sont:

- o Linéaire



- o Douce
- o Sans
- o Alpha

La transition 'Linéaire' fait décroître linéairement l'intensité des pixels d'une image au fur et à mesure que l'on avance sur l'autre.

La transition 'Douce' effectue une transition du même type mais à l'aide d'une fonction sinusoïdale.

La transition 'Sans' n'effectue aucune variation des valeurs des pixels donc pas d'atténuation.

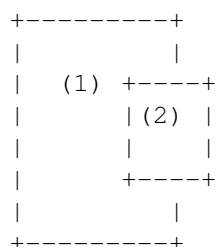
La transition 'Alpha' permet l'intervention d'un 'Alpha Channel', c'est-à-dire qu'une troisième image va piloter le taux de mélange de l'image. l'alpha channel doit être de même dimension que la deuxième image.

Remarque 1: Il est conseillé d'appliquer la même méthode de transition aux deux images, notamment dans le cas de constitution d'une mosaïque.

Remarque 2: La ~méthode~de~composition~ est appliquée après traitement des pixels par la méthode de transition décrite ici.

Deux cas typiques peuvent se présenter:

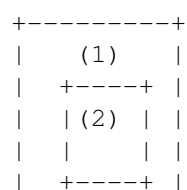
Cas 1  
~~~~~



Dans ce cas la transition déterminera :

- o L'atténuation de l'image (1) de puis le bord gauche de l'image (2) vers le bord droit de l'image (1) commun avec l'image (2).
- o L'atténuation de l'image (2) depuis le bord droit de l'image (1) commun avec l'image (2) vers le bord gauche de l'image (2).

Cas 2
~~~~~



```

      |           |
      +-----+

```

Dans ce cas la transition déterminera :

- o L'atténuation de l'image (1) de puis les bords de l'image (2) vers le centre de l'image (2).
- o L'atténuation de l'image (2) depuis son centre vers ses bords.

NOTE

Dans le cas de la transition par alpha channel, il n'y a pas de distinction entre ces deux cas.

--

## 1.115 Couche suivante

```

Couche Suivante / Couche Précédente
=====

```

Si l'image affichée fait partie d'une image multicouche ces deux options permettent alors de:

- o 'Naviguer' à travers les différentes couches de l'image active si l'on est en ~mode~Source~ 'Multicouche'.

ou

- o 'Naviguer' à travers les différentes couches de l'image à laquelle appartient la couche visualisée si l'on est en mode Source 'MonoCouche'.

--

## 1.116 Démarrer animation

```

Démarrer animation
=====

```

Cette 'animation' met en jeu seulement les couches de l'image active.

Pour stocker les différentes images graphiques, Lynx utilise la mémoire CHIP, il est donc nécessaire d'en disposer d'une réserve suffisante.

D'autre part pour obtenir une animation fluide il est conseillé d'ouvrir initialement un écran de visualisation comportant le moins de plans de bits possibles.

Lorsque qu'une image (couche) est dessinée pour la première fois ou lorsque que l'on ne dispose plus suffisamment de mémoire CHIP, l'image est dessinée de façon 'classique', car aucune bitmap pour la stocker n'existe déjà.

Pour stopper une animation en cours, il suffit de presser n'importe quelle touche du clavier.

--

## 1.117 Vitesse d'animation

Vitesse d'animation  
=====

Permet de régler la vitesse d'une animation, c'est-à-dire le nombre d'images à afficher par seconde.

Bien sur pour de grande images et/ou des écrans comportant beaucoup de plans de bits la vitesse maximale peut-être assez faible.

--

## 1.118 Libère BitMaps

Libère BitMaps  
=====

Quand une animation à été jouées ou lorsque l'on utilise les options de déplacement à travers les couches (Couche Précédente ou Couche Suivante), Lynx stocke les contenus successifs de la fenêtre de visualisation d'image en mémoire CHIP, et celle-ci peut alors assez rapidement être pleine.

Cette option permet alors de libérer la mémoire CHIP qui était allouée pour stocker les images.

--

## 1.119 HELP !

HELP  
=====

Invoque l'aide en ligne.

Pour que l'aide fonctionne il faut que le fichier 'Lynx.guide' figure dans le répertoire 'Lynx:'

--

---

## 1.120 Sortir

```
Sortir
=====
```

Détruit l'écran de visualisation. On se retrouve sous l'écran de contrôle.

Remarque: Si l'on a effectué une ~mesure~d'angle~ ou une ~mesure~de~longueur~ alors les paramètres déduits de ces mesures seront proposés pour effectuer une rotation ou une translation.

--

## 1.121 Editer

```
Editer la palette
=====
```

Ouvre l'éditeur de palette. Toutes les couleurs de la palette sont modifiables.

--

## 1.122 Charger

```
Charger Palette
=====
```

Permet de charger une palette de couleurs à appliquer à l'écran de visualisation.

Suivant le nombre de couleurs de l'écran de visualisation on accède à un jeu de palettes spécifique:

```
PAL4 -> Ecran 16 couleurs
PAL5 -> Ecran 32 couleurs
PAL6 -> Ecran 64 couleurs
PAL7 -> Ecran 128 couleurs
PAL8 -> Ecran 256 couleurs
```

Il est donc vivement déconseiller d'ouvrir une palette incompatible avec l'écran de visualisation ouvert.

--

## 1.123 Sauver

---

### Sauvegarder Palette =====

Permet de sauver l'actuelle palette de couleur de l'écran de visualisation

La requête de fichier ouverte est positionnée sur le répertoire correspondant au nombre de couleurs de l'écran de visualisation ouvert.

Il est vivement déconseillé de sauvegarder la palette ailleurs que dans ce répertoire.

--

## 1.124 Information

### Information =====

Renvoie des information à propos de la zone image sélectionnée (voir ~Sélection~):

- o Largeur et hauteur de la zone en pixels.
- o Barycentre de la zone en pixel
- o Somme des intensités de tous les pixels situés dans le rectangle.
- o Un gadget seuil permet de fixer l'intensité minimale qu'un pixel doit avoir pour être pris en compte dans la détermination du barycentre et de la somme (utile en astronomie pour mesurer le barycentre d'une étoile sans considérer le fond de ciel).

--

## 1.125 Définir Masque

### Définir Masque =====

Permet de restreindre à la zone sélectionnée tout traitement ultérieur dans le domaine spatial.

Il faut en outre mettre le mode de masquage sur ON.

--

---

## 1.126 Extraire

Extraire  
=====

Crée une nouvelle image à partir de la sélection.

En mode de visualisation en trichromie, une image composée de trois couches est créée.

L'implantation de cette fonction n'est pas terminée : Si l'on extrait une sélection à partir d'une visualisation en Pseudo-Couleur et que le mode Source est PileImage une seule couche est créée.

--

## 1.127 Abandonner

Abandonner  
=====

Efface la sélection. Aucun des opération de sélection n'est possible jusqu'à la création d'une nouvelle sélection à l'aide de l'outil de ~sélection~.

--

## 1.128 Histogramme

Histogramme  
=====

Ouvre une fenêtre pour y tracer l'histogramme des pixels de la zone de sélection.

On trouvera en limite des abscisses de l'histogramme des valeurs la plus petite et la plus grande parmi les pixels de la zone de sélection.

L'échelle des ordonnées est logarithmique afin de distinguer aussi bien les population fortement représentées que celles faiblement présentes.

--

## 1.129 Tout Effacer

```
Tout effacer
=====
```

Efface toutes les marques apposées (voir outil ~Synthèse~) sur la fenêtre de visualisation destinées à la ~génération~d'une~synthèse~.

--

## 1.130 Générer Image

```
Générer image
=====
```

Calcule une image à partir des points de mesures (croix) posées sur la fenêtre de visualisation grâce à l'outil ~Synthèse~.

Cette possibilité est surtout intéressante en astronomie pour effectuer une synthèse du fond de ciel d'une image (en vue de son élimination) dans les cas où cette information n'est pas déjà disponible.

On aura recours à cette technique pour éliminer l'effet de la Lune, par exemple, qui va polluer une image avec un fond de ciel présentant un dégradé, donc une non uniformité.

On pourra donc, après soustraction à l'image originale du fond ainsi synthétisé obtenir ce que l'on aurait eu sans l'objet polluant (Lune, étoile brillante...).

On applique aussi cette technique lorsque l'on n'a pas de carte de sensibilité de la caméra CCD qui a servi à faire l'image: on soustrait également le fond synthétisé.

Une fois le calcul terminé, l'image active est celle représentant la synthèse ainsi réalisée.

--

## 1.131 ToolTypes

```
ToolsTypes (Types d'outils)
=====
```

Lynx reconnaît actuellement deux Types d'outils:

```
SCREEN=<CUSTOM | PUBLIC>
```

```
PRIORITY=<priorité>
```

Le type d'outil SCREEN permet de choisir si l'on veut avoir l'écran de contrôle de Lynx sur un nouvel écran (CUSTOM) ou bien sur l'écran public par défaut (PUBLIC).

Le type d'outil PRIORITY sert à fixer la priorité de Lynx (par défaut 0).

--

## 1.132 Correction pixel par pixel

Corrections pixel par pixel  
=====

La correction pixel par pixel permet l'analyse fine et la correction de pixels individuels de l'image.

L'option ouvre une nouvelle fenêtre comportant 9 cellules représentant les 9 valeurs des pixels du pavé centré sur le pixel pointé par la souris.

Pour actualiser les cellules il suffit de promener la souris sur l'image en maintenant la touche 'SHIFT' enfoncée. Les 9 cellules seront mises à jour au fur et à mesure du déplacement de la souris sur l'image.

La cellule centrale peut recevoir une nouvelle valeur entrée par l'utilisateur et qui sera immédiatement affectée au pixel qui était pointé par la souris

Sous les 9 cellules de valeurs il y a deux boutons:

- 1 - 'Moyenne'. Cliquer sur ce bouton a pour effet de remplacer la valeur du pixel central par la moyenne de ses 8 voisins.
- 2 - 'Annuler'. Annule toutes les modifications faites sur le pixel central depuis la dernière mise à jour des 9 cellules de valeurs.

REMARQUE: Utilisé conjointement avec le ~zoom~ le pointage de pixels est plus précis !

--

## 1.133 Fonte...

Fonte...  
=====

Permet de choisir la fonte qui sera utilisée pour la pose d'~annotations~ sur l'image.

---



--

## 1.134 Défaire

Défaire  
=====

Inverse la dernière annotation placée sur l'image, que ce soit un texte ou un graphique.

Voir aussi `~Graphismes~d'annotation~`.

--

## 1.135 Effacer tous les graphiques

Effacer tous les graphiques  
=====

Efface de l'image toutes les annotations réalisées depuis la dernière `~intégration~dans~l'image~`.

Voir aussi `~Graphismes~d'annotation~`.

--

## 1.136 Intégrer dans l'image

Intégrer dans l'image  
=====

Intègre effectivement les annotations réalisées dans l'image. Une fois l'intégration faite il n'est plus possible de venir en arrière, on ne peut le faire qu'avant avec `~Effacer~tous~les~graphiques~`.

Voir aussi `~Graphismes~d'annotation~`.

--

## 1.137 Graphiques

## Graphismes d'annotation =====

Les graphismes d'annotation permettent d'apposer sur une image des informations permettant de préciser le contenu de celle-ci ou de porter l'attention sur certains détails.

Il ne s'agit pas d'un module de dessin. En effet la couleur des tracés ne peut être choisie et de plus la fonction n'est pour l'instant opérationnelle qu'au niveau d'une couche d'image.

--

### 1.138 Exemples pratiques

#### Exemples pratiques =====

```
1 ~Faire~une~image~couleur~avec~trois~images~monochromes~
2 ~Faire~varier~contraste~et~luminosité~d'une~image~~~~~
3 ~Réaliser~un~effet~du~style~'peinture~à~l'huile'~~~~~
4 ~Réaliser~une~'solarisation'~~~~~
5 ~Mise~en~orbite~autour~de~Saturne~!~~~~~
6 ~Eliminer~le~bruit~présent~dans~une~image~~~~~
7 ~Réaliser~un~'masque~flou'~~~~~
8 ~Réaliser~une~convolution~dans~le~domaine~de~Fourier~~~
9 ~Réaliser~une~déconvolution~dans~le~domaine~de~Fourier~
10 ~Un~peu~de~reconnaissance~de~formes...~~~~~
11 ~Décomposition~en~ondelettes~~~~~
```

#### NOTE

La version requise indiqué au début de chaque exemple n'est pas obligatoire mais est suffisante (version 8bits) ou plus adaptée (version 16bits) libre à vous de ne pas suivre ce conseil.

--

### 1.139 Faire une image couleur avec trois images monochromes

#### Faire une image couleur avec 3 images monochromes =====

Le but du jeu est de réunir dans la même pile d'image les trois images monochromes.

NOTE: L'ordre de chargement des images (rouge, vert ou bleu) n'a pas d'importance.

1. Placer le mode Destination sur 'NouvPile'.
2. Charger la première image.
3. Placer le mode Destination sur 'NouvCouche'.
4. Charger les deux autres images.

Lors de la première visualisation, Lynx vous demandera de préciser quels sont les canaux rouge, vert et bleu.

--

## 1.140 Faire varier contraste et luminosité d'une image

Faire varier contraste et luminosité d'une image  
=====

Plusieurs méthodes sont possibles:

1. Changer les seuils de visualisation.  
Rappelons que ces seuils spécifiques pour chaque couche d'une image  
On peut ainsi renforcer ou atténuer l'importance d'une composante (rouge, vert ou bleu) par rapport aux autres.
2. Effectuer un transcodage (menu VISUALISATION)  
Grâce à la composition de 3 tables simples il est possible de réaliser un grand nombre d'effets (16777216 pour chaque triplet !).  
Les fonctions 'Contraste+' et 'Contraste-' agissent directement sur le paramètre <contraste> de l'image. La fonction Logarithme élève les faibles lumières sans pour autant saturer les fortes intensités, on agit ainsi sur le paramètre <lumière>.

La fonction de transcodage ainsi créée s'applique à toutes les images (couches confondues).

--

## 1.141 Réaliser un effet du style 'peinture à l'huile'.

Réaliser un effet du style 'peinture à l'huile'  
=====

Version requise: 8bits

---

Lynx ne possède que très peu d'opérateurs d'effets spéciaux. En fait vous l'aurez remarquer, les opérateurs disponibles sont pour la plupart des opérateurs de bas niveaux ne réalisant que des opérations très simples.

En fait il suffit de les combiner pour obtenir certains effets assez complexes.

L'effet de 'peinture à l'huile', disponible sur la pluparts des logiciels de retouche d'images est facilement reproductible avec ce logiciel. Foivi la procédure à suivre:

1. Régler correctement le Mode de Traitement .
2. S'assurer que l'image est correctement normalisée. Seuils se visuali-  
correspondant bien à 0 pour le seuil bas et 255 (32767 en v16bits)  
pour le seuil haut.  
Sinon normaliser l'image à 0 - 255 (v8bits) ou 0 - 32767 (v16bits)  
à l'aide de la fonction ~Normaliser~.
3. Cliquer sur ~Transcodage~ (menu TRAITEMENT SPATIAL) et sélectionner  
la fonction 'ET logique'. Entrer comme coefficient 224 (Vers. 8bits)  
ou 28896 (Vers. 16bits). Ceci aura pour effet de discrétiser  
fortement les niveaux de lumière qui ne seront plus codés que sur 3  
bits ( $224 = 128 + 64 + 32$ ).
4. Cliquer sur ~Flt~non~lin~ et choisissez le filtre 'médian' sans  
modifier sa structure. Entrer ensuite 100 comme coefficient d'effica-  
cité.
5. A l'aide de la fonction ~MultConst~ multiplier l'image ainsi obtenue  
par 1.138393, qui est le rapport  $255 / 224$ .
6. Vous pouvez visualiser le résultat !

Vous pouvez bien sur jouer sur les différents paramètres tels que:

- o Valeur pour le transcodage: 192, 240 sont d'autres valeurs utilisables  
Le facteur multiplicatif sera alors (respectivement) 1.328125, 1.0625.
- o La géométrie du filtre médian.

--

## 1.142 Réaliser une solarisation

Réaliser une 'solarisation'  
=====

Rien de plus simple !

1. Cliquer sur ~Transcodage~ (menu AFFICHAGE), sélectionner la table de  
transcodage 'Solarisation' et fixer la réglette vers la valeur 100.

Choisissez Ok.

2. C'est fait ! le temps de traitement est égal à 0.00s !!

Cela reste évidemment valable quelque soit la table de transcodage utilisée et cela montre bien la puissance de ce programme !

Toutes les images que vous visualiserez ensuite seront en apparence solarisées.

Si vous voulez ensuite sauvegarder l'image sous son aspect solarisé il suffit d'effectuer une sauvegarde au ~format~iff~, depuis l'écran de contrôle ou l'écran de visualisation.

Voir aussi ~Sauver~sous...~IFF~image~ ~Sauver~fenêtre~

--

## 1.143 Mise en orbite autour de Saturne !

Mise en orbite autour de Saturne !  
=====

Version requise: 8bits

Ce petit exemple est destiné à vous montrer les capacités de Lynx en matière de traitements 'artistiques'.

- 1 - Chargez les images 'Lake.iff' et l'image 'Saturne.iff'.
  - 2 - Sélectionnez ensuite l'image 'Lake'.
  - 3 - Cliquez sur ~Pseudo-Coul~ et choisissez une des 3 composantes.
  - 4 - Après affichage de cette composante, choisissez l'option ~Composition~ du menu 'Projet'. Choisissez alors l'une des trois composantes de l'image 'Saturne'.
  - 5 - Placer la composante de 'Saturne' de façon à ce quelle se superpose sur le ciel de l'image de fond. Pour déplacer l'image de Saturne, je rappelle qu'il suffit d'attraper le haut de son image (qui cache une barre de déplacement).
  - 6 - Cliquez avec le bouton droit de la souris. La fenêtre d'options de composition apparait. Choisissez alors 'Addition' dans le menu 'Composition' et 'Alpha' pour les 'transition' des deux images.
  - 7 - Comme vous avez choisi l'option de transition 'Alpha', on vous demande alors de choisir l'image qui va servir d'alpha channel. Choisissez alors la composante rouge de saturne.
  - 8 - Visualisez ensuite le résultat à l'aide d'une visualisation en trichromie.
-

--

## 1.144 Eliminer le bruit présent dans l'image

Eliminer le bruit présent dans une image  
=====

Il peut arriver qu'une image soit bruitée. Ce bruit se caractérise souvent par des pixels isolés d'intensités très différentes de leur voisinage.

Deux méthodes sont envisageables :

1. Filtrer l'image par un filtre passe-bas de faible puissance.  
Ceci permettra d'éliminer relativement efficacement les pixels représentatifs du bruit tout en conservant l'information significative contenue dans l'image.
2. Filtrer l'image par un filtre statistique, le filtre médian, qui a pour effet d'éliminer seulement les pixels aberrants en fonction de leur voisinage si l'on choisit correctement son facteur d'efficacité.

La première méthode est assez rapide mais présente un inconvénient majeur : tous les pixels de l'image subiront l'effet du filtre si modéré soit-il. Le résultat est que le bruit parasite aura disparu en même temps que les plus fins détails de l'image, qui, eux, n'étaient pas du bruit, cela peut dans certains cas être assez néfaste.

La deuxième méthode, plus coûteuse en temps de calcul, est cependant plus efficace et altère moins l'image.

Résumons ce qu'il faut faire en utilisant chacune des deux méthodes :

Méthode par le filtrage passe-bas

1. Sélectionnez l'image à traiter puis Cliquez sur 'Convolution'.
2. Choisissez un filtre 'passe-bas faible' et choisissez 'Ok'.
3. Vous pouvez regarder le résultat ! S'il ne vous satisfait pas réessayez avec d'autres filtres passe-bas.

Méthode par le filtre médian

1. Sélectionnez l'image à traiter puis Cliquez 'Filt-nlin'.
2. Choisissez le filtre 'médian'. Normalement la forme du filtre (un pavé 3x3) est bien adaptée, cependant si le bruit occupe des zones de plusieurs pixels de surface, il serait bon d'élargir la forme du filtre médian à 5x5, mais dans ce cas le temps de traitement va augmenter considérablement !

3. Choisissez 'Ok' puis entrer une valeur faible pour l'efficacité du filtre: de l'ordre de 10 à 20, ceci pour n'éliminer que les pixels vraiment représentatifs du bruit.
4. Regarder le résultat. Si il reste quelques pixels isolés vous pouvez réessayer en augmentant légèrement l'efficacité du filtre ou en passant à 5x5 la taille de la matrice s'il reste de gros pavés de bruits.

--

## 1.145 Réaliser un 'masque flou'

Réaliser un 'masque flou'  
=====

Cet exemple vous montrera comment améliorer la qualité d'une image en faisant ressortir ses plus fins détails.

Le principe du masque flou est bien connu des photographes qui l'utilisent pour accentuer les détails noyés dans de forts gradients de densité du film. La méthode consiste à réaliser sur une émulsion une copie négative du cliché original avec un flou de pénombre judicieusement choisi. Après traitement, le masque ainsi obtenu en placé en coïncidence sur l'original et l'ensemble est copié. Les détails sont inexistant dans le masque mais en revanche celui-ci conserve les larges gradients de lumière qui annihilent ceux du négatif original. En conséquence, le résultat final montre un rendu des détails considérablement amélioré.

Une démarche analogue est adoptée pour le traitement numérique:

L'image à traiter est lissée par un filtre de type passe-bas puis on lui soustrait le masque flou ainsi obtenu. Le résultat est une image où les hautes fréquences, qui caractérisent les variations les plus rapides, dominent par suppression des variations lentes d'intensité.

Le masque flou est en fait un filtre passe-haut très puissant. Si puissant même qu'en pratique on modérera son ardeur en ajoutant un peu de basses fréquences afin de donner un aspect plus agréable à l'image finale.

Pratiquement on réalisera les opérations suivantes:

1. S'assurer que l'image est correctement normalisée. Seuls se visualisant correspondant bien à 0 pour le seuil bas et 255 (32767 en v16bits) pour le seuil haut.  
Sinon normaliser l'image à 0 - 255 (v8bits) ou 0 - 32767 (v16bits) à l'aide de la fonction ~Normaliser~.
2. Fixer le mode de traitement suivant que l'on a à traiter une image multicouche ou monocouche et le mode de Destination dans ce dernier cas.
3. Filtrer l'image par un filtre passe-bas:
  - Soit avec une matrice de convolution

- Soit directement avec l'opérateur 'Gaussienne'.  
Prenez par exemple la matrice de convolution 'Moyenne 3x3'.

4. Sélectionner l'image originale.
5. Lui soustraire le masque flou créé en (3).
6. Version 8 bits uniquement:  
Translater l'histogramme de l'image ainsi créée à l'aide de la fonction ~ADDconstante~: Valeur -64 et 'Même Image'
7. Diviser par 2 (en lumière) l'image originale par la fonction ~Multconst~. Constante = 0.5 et cliquez sur 'Même Image'.
8. Additionner le résultat avec l'image créée en (5 ou 6) en réalisant une ~Addition~ simple.
9. Ajuster les seuils de visualisation:  
Version 8bits:   seuil min <- 64  
                  seuil max <- 192  
Version 16bits: seuil min <- 0  
                  seuil max <- 16384
10. Visualiser le résultat.

--

## 1.146 Réaliser une convolution dans le domaine de Fourier

Réaliser une convolution dans le domaine de Fourier  
=====

Version requise: 16bits

Tout d'abord pour tout ce qui est transformée de Fourier il est préférable (mais non indispensable) de travailler avec la version 16bits du programme.

1. Charger l'image 'M51.lynx'
2. Réaliser le ~Transcodage~ (menu TRAITEMENT SPATIAL) suivant:  
'Troncature'  
avec Seuil Min = -32767 (V16bits) ou 0 (V8Bits)  
      Seuil Max = 0  
Ceci a pour effet de créer une nouvelle image de même dimension que l'originale mais remplie de 0.
3. Visualiser Cette nouvelle image en cliquant sur ~Pseudo-Coul~.  
Sélectionner le gadget 'A' de la boîte à outils.  
Choisissez la fonte que vous désirez pour écrire: menu 'Graphiques'  
~Fonte...~.  
Enfin cliquer dans l'image pour entrer un texte. Entrez simplement



un 'A' et placez-le au centre de l'image.  
Sélectionnez ensuite le'option de menu 'Intégrer les graphiques'  
dans le menu 'graphiques'.  
Sortir ensuite de l'écran de visualisation: Menu 'Projet' 'Sortir'.

4. Effectuer la transformée de Fourier de cette image:  
Bouton ~TFR~directe~. Visualisez le résultat.
5. Sélectionnez l'image originale (celle chargée en 1.) et effectuer  
sa transformée de Fourier.  
Visualisez le résultat.
6. Cliquez sur ~Convolution~ (menu TRAITEMENT FREQUENTIEL) et  
Choisissez l'autre image.  
Visualisez le résultat.
7. Cliquer sur ~TFR~inverse~.  
A la question "Normalise image ?" répondez OUI.  
A la question "Dois-je faire une permutation ?" répondez OUI.
8. Avant de visualiser le résultat. Cliquez sur ~Transcodage~ (menu  
AFFICHAGE) et sélectionnez la table 'logarithme'.  
Cliquez ensuite sur ~Histogramme~ (menu VISUALISATION) et fixer  
le seuil bas à 0 et de seuil haut à peu près au trois-quarts  
de la dynamique de l'histogramme.
9. Visualisez. Vous devez voir alors des 'A' partout là où il y  
avait des étoiles. De même la galaxie M51 ne semble plus être  
qu'un amas de 'A'.

Vous pouvez ensuite essayer un masque flou, mais vous n'aurez jamais qu'un  
tas de 'A' plus nets !

Le seul moyen de se débarrasser de ces 'A' pour retrouver l'image originale  
est la déconvolution dans le domaine de Fourier; ceci fait l'objet de  
l'exemple~suivant.

--

## 1.147 Réaliser une déconvolution dans le domaine de Fourier

Réaliser une déconvolution dans le domaine de Fourier  
=====

Version requise: 16bits

Si vous n'avez pas suivi l'exemple~précédent reportez-vous y avant  
d'attaquer celui-ci.

Reprenez l'image résultante du traitement de l'exemple précédent; on a donc  
un tas de 'A' en forme de galaxie !

Reprenez aussi l'image de la réponse impulsionnelle (le 'A' isole au centre d'une image vide), et si vous ne l'avez plus recréer-la. Ensuite SAUVEGARDER la car vous en aurez besoin pour chaque tentative.

Les différentes étapes sont les suivantes:

1. Si vous venez de l'étape (5) Calculer la TF (bouton ~TF~directe~) de l'image de la réponse impulsionnelle, sinon calculer les TF des 2 images décrites ci-dessus.
2. Sélectionner la TF de la galaxie perturbée (!).
3. Cliquer sur ~Déconvolution~. A ce moment choisissez une valeur pour la constante de Wiener (la valeur par défaut peut convenir pour un premier essai). En fait une valeur nulle est ici optimal vu que l'on connaît parfaitement la réponse impulsionnelle. Il faudra essayer en utilisant un 'A' un peu différent de celui utilisé pour la convolution.
4. Ajuster les seuils de visualisation de l'image du spectre ainsi obtenu à l'aide de l'histogramme~ et visualiser le résultat. Un spectre correct doit montrer une nette prédominance des basses fréquences. Si il y a de forte valeur sur les bord de l'image (hautes fréquences) alors il est fort probable que le choix sur la valeur de la constante de Wiener était incorrect.
5. Cliquer sur ~TFR~inverse~ pour revenir dans le domaine spatial, ajuster les seuils de visualisation et regarder le résultat. Si celui-ci semble très éloigné de l'image d'origine alors reprenez à partir de l'étape 1.

--

## 1.148 Un peu de reconnaissance de formes...

Un peu de reconnaissance de formes...  
=====

Version requise: 16bits

Cet exemple, un peu plus complexe, vous offrira une première approche de la reconnaissance de forme au moyen de la transformation de Fourier.

Charger l'image Texte.iff (Ouvrir~sous...~Format~IFF).

L'objectif va être d'isoler une l ou un ensemble de lettre et d'essayer de les retrouver dans ce texte.

La méthode consistera à isoler le motif recherché dans notre texte et de le mettre au centre d'une nouvelle image vide de même dimension que l'image du texte. On réalisera ensuite la corrélation entre le texte et le motif ainsi extrait.

1. Visualiser l'image 'Texte'. Sélectionner l'outil~de~sélection~ d'une zone d'image. Sélectionner ensuite l'option de menu 'Extraire' du menu 'Sélection'.
2. ~Redimensionner~ l'image à 256x256 (ces valeurs doivent correspondre aux valeurs par défaut). Renommez-la et appelez la 'Motif'.
3. Sauver cette image ou faites-en une copie.
4. Calculer les TF des 2 images (le texte et le motif extrait).
5. Sélectionner le spectre du texte ('Texte-F') et cliquez sur 'Corrélation'.
6. Cliquez sur ~TFR~inverse~ pour obtenir l'image de la corrélation entre le texte et le motif recherché.
7. Visualisez l'image ainsi obtenue. la valeur de chaque pixel représente la probabilité de trouver à cet endroit le motif recherché, les valeurs correspondant à la valeur maximale autorisée (255 ou 32767) indiquent une probabilité de 100%. Quittez ensuite l'écran de visualisation
8. Effectuer maintenant un ~transcodage~ (menu TRAITEMENT SPATIAL) et sélectionnez la table 'Linéaire'. Fixez les niveaux d'entrée et de sortie aux valeurs suivantes:

|           | Vers 8bits | Vers 16 bits |
|-----------|------------|--------------|
| entrée:   |            |              |
| minimum : | 254        | 32766        |
| maximum : | 255        | 32767        |
| sortie:   |            |              |
| minimum : | 0          | 0            |
| maximum : | 255        | 32767        |

L'image obtenue ne contient donc que quelques pixels, ceux-ci indiquant avec une confiance absolue la position de chaque motif identique au motif recherché.

8. Renommez l'image ainsi obtenue et appelez-la 'Corrélation'.
9. Sélectionner l'image 'Motif' et calculez sa transformée de Fourier (bouton ~TFR~directe~). Sélectionnez l'image 'Corrélation' et calculez de même sa TF.
10. Sélectionnez le spectre de 'Corrélation': 'Corrélation-F'. Cliquez sur ~Convolution~ et sélectionnez 'Motif-F'. On va effectuer ainsi la convolution de l'image nettoyée de la corrélation (où il ne reste que les pics correspondant au motif recherché) avec le motif lui même.  
A la question "Normalise Image ?" répondre OUI  
A la question "dois-je faire une permatution ?" répondre OUI
11. Ajuster les seuils de visualisation et regardez le résultat on doit voir, à la place de chaque pic dans l'image 'Corrélation', une reproduction du motif recherché.

12. Recharger ensuite l'image 'Texte' et lui soustraire l'image obtenue en (11). Effectuer ensuite une ~normalisation~. Le résultat doit être le texte avec un trou partout où le motif était présent.

--

## 1.149 Approche de l'analyse en ondelettes

Décomposition en ondelettes  
=====

Version requise: 16bits

La décomposition en ondelette est quelque chose de très récent dans le domaine du traitement des images. Cela fait appel, une fois de plus, à la transformation de Fourier.

Le principe de cette analyse est d'extraire de l'image ses principales caractéristiques en terme d'échelle de grandeur... les plus fins détails représentent l'échelle la plus petite d'une image, tandis que les plus grandes structures représentent les grandes échelles.

Nous allons prendre un exemple. Chargez l'image 'Messier51.lynx'.

Sur cette image on voit plusieurs choses:

- o Une galaxie
- o Les bras de cette galaxie
- o Des étoiles brillantes
- o Des étoiles faibles
- o Du bruit

Grâce à la décomposition en ondelettes nous sommes capable d'isoler chacun de ces éléments fondamentaux de cette image.

**NOTE IMPORTANTE:** Lors des opérations de transformation de Fourier inverse on vous demande si l'image doit être normalisée, la réponse est OUI avec la version 8bits et indifférente avec la version 16bits.  
De plus on vous demande aussi si l'on doit faire une permutation des cadrans de l'image, la réponse est NON dans tous les cas.

Voici comment procéder:

1. Sélectionner l'image 'Messier51' et cliquer sur ~TFR~directe~ pour calculer sa transformée de Fourier.
2. Cliquer sur ~Filtrage~ et choisissez le filtre 'Wavelet': il s'agit du filtre magique qui effectue l'extraction des fréquences représen-

tatives d'une échelle de grandeur.

Amener la réglette sur la valeur 1. Cette valeur, la plus petite exploitable, isole les plus hautes fréquences, donc les plus fins détails, donc la plus petite échelle de grandeur.

3. Cliquer sur ~TFR~inverse~ pour revenir au domaine spatial. Ajustez les seuils de visualisation. l'histogramme de l'image va présenter un pic, il est préférable d'ajuster le seuil bas dessus et le seuil haut de façon à obtenir une image suffisamment lumineuse (sans saturation cependant).  
Que voit-on ? Des étoiles, quasiment plus de galaxie et aussi du bruit; en effet le bruit se manifestant surtout vers les hautes fréquences, il est normal de le voir ressortir ici. On a donc extrait les structures les plus fines de l'image.
4. Recharger 'Messier51.lynx' et répéter l'opération plusieurs fois à partir de (2) avec pour comme valeurs successives pour le filtre Wavelet 2, 4, 8, 16, 32, 64, 128, 256 (ca va prendre du temps !).  
A chaque fois on multiplie par deux, en fait cette valeur correspond au FACTEUR d'ECHELLE désiré, et, en multipliant par deux sa valeur, on obtient des structures 2 fois plus importantes. En fait tout se passe comme si on s'éloignait de cette image en doublant à chaque étape la distance qui nous en sépare et que l'on cherche les plus fins détails observables, le résultat serait alors strictement le même.

Prenez l'image réalisée avec la valeur 4 du filtre Wavelet, cela correspond au plus fins détails qui aurait été vus si l'on avait été 4 fois plus éloigné de l'objet photographié (car  $4/1 = 4$ ). Si vous n'êtes pas convaincu réalisez le filtrage par ondelettes (valeur du filtre Wavelet = 1) avec l'image Messier51.lynx en l'ayant réduite d'un facteur 4 (Bouton ~Echelle~ avec comme valeurs 0.25 en x et y) et ensuite agrandissement d'un facteur 4 du résultat. On doit trouver une image fort ressemblante à celle réalisée avec l'image originale sur laquelle on a appliqué le filtre Wavelet avec pour valeur 4.

La première application est bien sur d'extraire un des éléments fondamentaux d'une image sans être parturbé par d'autres détails. Il est ainsi facile d'extraire les étoiles de notre image 'Messier51.lynx' en dehors de la galaxie elle-même.

Une autre application de ce traitement est aussi pouvoir ainsi renforcer certains détails ou certaines 'classes' de détails d'une image. Supposons que vous ayez l'intention de renforcer la structure spirale de l'image de la galaxie Messier51, il suffit alors de réaliser le filtrage au moyen du filtre Wavelet avec pour valeur 4 (qui isole bien la structure spirale) et d'additionner ensuite ceci à l'image originale de Messier51. Vous verrez alors une forte amélioration du contraste au niveau de la structure spirale.

Attention toutefois à ne pas trop abuser de ce moyen, car vous risqueriez de voir alors rapidement apparaître des valeurs négatives dans l'images. En fait le principe du masque flou est préférable et surtout plus simple s'il l'on désire renfoecer les contrastes d'une image.

--

## 1.150 Fonction non implémentée

La fonction n'est pas encore implémentée. Désolé !

--

## 1.151 Historique

Program history  
=====

|                |          |                                                                                                   |
|----------------|----------|---------------------------------------------------------------------------------------------------|
| Summer 1993    | V0.0     | : Start of the development of 'Lynx'.                                                             |
| October, 1993  | V0.5     | : Most of the image processes implanted<br>First version of the graphical interface.              |
| November, 1993 | V0.6     | : Fast Fourier Transform implanted !!<br>First On-Line help. Not very great. Quickly<br>canceled. |
| Januar, 1994   | V0.7     | : Visualisation of color images.<br>Reading of IFF files.                                         |
| Februar, 1994  | V0.8     | : Firsts Multichannels processes.<br>Reading and writing of some stranges<br>image formats.       |
| May, 1994      | V0.9     | : ARexx port implanted.<br>3.0+ compatible? Use of 3.0 functions for<br>visualisation.            |
| April, 1995    | V1.00a_I | : First alpha-version.                                                                            |
| 20/05/95       |          | : V1.0B first beta-release. French documen-<br>tation only. Except this page !                    |

--

## 1.152 A faire

Développements futurs  
=====

Lynx n'en est qu'à sa première beta-version, beaucoup de choses sont déjà réalisées mais beaucoup d'autres choses restent à faire (et seront faites) dont :

#### Visualisation

- o Dessin d'image en 3D.
- o Module de mesures de somme d'intensités (pour la photométrie par ex.)
- o Sélection de surfaces de formes quelconques.
- o Visualisation de plusieurs images en couleurs simultanément.
- o Choix d'un autre système de sélection des couleurs : à palette fixe.
- o Ajustement automatiques des seuils sur simple demande utilisateur.
- o Module de dessin ??

#### Traitement d'images dans le domaine spatial

- o Convolution directe avec des images (de petites tailles)
- o Restorations par les méthodes de:
  - Lucy-Richardson
  - Van-Cittert
- o Analyse par ondelettes.

#### Traitement morphologique des images

- o Erosion / Dilatation
- o Squeletisation

#### Opérations/Transformations diverses

- o Transformation colorimétrique étendue à d'autres systèmes tels que:
  - YUV
  - CMYK

#### Opérations géométriques

- o Réalisation d'un module complet de cartographie.

#### Traitement d'images dans le domaine de Fourier

- o Choix de la représentation de la TF:
  - Module (ce qui est fait actuellement)
  - Phase.
  - Partie réelle.
  - Partie imaginaire.
- o Filtrage par sélection en fonction de différents paramètres:
  - Intensité du module
  - Valeur de la phase
- o Reconstitution par la méthode du maximum d'entropie.
- o Restauration par déconvolutions itératives.

#### ARexx

- o Développement d'un grand nombre de fonctions, permettant d'effectuer TOUS les traitements disponibles.

#### Entrées/Sorties

- o Lecture d'un format binaire brute (Raw data).

#### Documentations

- o Doc en anglais (of course !)

#### Général

- o Possibilité d'écrire des modules externes (quelques tests réalisés avec succès sont prometteurs).
  - o Réécriture de la GUI avec une librairie objet (BGUI me semble meilleur du point de vue consommation\_de\_ressources/efficacité).
  - o Gestion d'un fichier de réglages par défaut.
-

--

## 1.153 Bogues

```
)-: Bogues :-(  
=====
```

'Bogues' est la traduction française de 'bugs'. Il s'agit donc des quelques petits problèmes de fonctionnement qui sont connus.

```
* : petit bogue pas méchant.  
** : bogue assez sérieux.  
***: bogue très méchant.
```

```
* Pour une visualisation en mode HAM il vaut mieux que le nombre de  
couleur du mode (sélectionné avec le mode d'écran) soit celui du  
nombre de couleurs de base, autrement les menu de l'écran de  
visualisation risque de baver !
```

```
* Toujours en mode HAM. Sur les machines OCS/ECS en mode HAM  
haute résolution les chiffres indiquant la valeurs des pixels  
(fenêtre 'info pixel') peuvent donner des résultats erronés.  
Cela est du au fait que l'on affiche qu'un seul pixel sur deux afin  
de respecter le ratio de l'image.
```

```
*** La version 68000 peut (cela n'est pas toujours le cas) se planter  
lorsque l'on déplace la réglette (slider) de la fenêtre de trans-  
codage appliquée à la visualisation. Ce bogue n'existe apparamment  
pas avec les autres versions.
```

```
* Le transcodage d'une image monochromatique en mode source 'PileImages'  
ne fonctionne pas, il faut passer en mode source 'MonoCouche' pour  
que cela fonctionne.
```

--

## 1.154 Stéphane Poirier

Lynx à été écrit par Stéphane Poirier

Stéphane Poirier  
20 avenue Claude Vellefaux  
75010 Paris

FRANCE

---



Tél: (+33) (1) 42 00 49 01

--

## 1.155 Enregistrement

Enregistrement  
=====

Pour s'enregistrer il suffit d'envoyer à l'~auteur~ la somme de:

150 Frs  
ou 40 DM  
ou US\$ 30

sous forme de cash, chèque (pour les utilisateurs résidents en France), ou eurochèques.

Il est également nécessaire de m'indiquer vos:

Nom  
Prénom  
Adresse complète

Type d'Amiga utilisé:  
Modèle  
Version du WorkBench  
Mémoire  
Processeur/Coprocesseur  
Carte graphique

Les utilisateurs enregistrés recevront alors la dernière version de Lynx, adaptée à leur configuration matérielle (CPU+FPU).

Ils recevront en outre toutes les futures mises à jours et corrections de bugs.

--

## 1.156 Copyright

Copyright  
=====

Lynx est copyright © 1993-1995 Stéphane Poirier

Lynx est shareware. L'auteur est propriétaire du copyright de ce logiciel.  
Les personnes désirant utiliser sérieusement Lynx doivent s'enregistrer

---

auprès de l'auteur.

La distribution de la version non-enregistrée est permise tant qu'aucun profit de nature commerciale n'en est tiré. Le prix d'une disquette contenant la version non-enregistrée de Lynx ne doit en aucun cas excéder la somme de 15 Francs (ou US\$3 ou 5 DM). L'inclusion de la version non-enregistrée dans une collection de logiciel domaine public est autorisée si elle ne va pas à l'encontre des dispositions énumérées ci-avant. Vous ne devez modifier, ou ajouter, ou détruire, aucun des fichiers inclus dans cette archive. Elle doit restée complète et inchangée.

L'auteur ne peut garantir le fonctionnement correct de Lynx et ne peut être tenu pour responsable d'éventuelles conséquences néfastes de l'utilisation de Lynx. Les mises à jour et corrections ne sont pas garanties.

Lynx utilise la `reqtools.library`, non fournie avec l'archive. La `reqtools.library` est copyright © François Nico.

Lynx utilise l'`iff.library`, incluse dans l'archive. L'`iff.library` est copyright Christian A. Weber.

#### IMPORTANT

Pour le moment la version gratuite n'est pas bridée et ne le sera pas tant que ce programme sera en version bêta.

--

## 1.157 Remerciements

Remerciements  
=====

Je remercie François Nico, auteur de la `reqtools.library` ainsi que Christian A. Weber auteur de l'`iff.library`, qui, grâce à leur travail, ont fourni de puissants outils de développement.

Je remercie les différentes personnes qui ont permis, par leur tests intensifs, de déboguer Lynx, à savoir:

Etienne Vogt  
Ludovic Meynadier  
Pierre Bernardo  
Arnaud Leclerc  
Gaël Marziou

--

## 1.158 Index

---

## Index

=====

## Documents

A~faire  
A~propos~de~Lynx...  
Abandonner  
AddConstante  
AddImage  
Addition  
AddrCurrentImage  
Affichage  
ARexx  
Boîte~à~outils  
Bogues  
Buffers~d' images  
Calcul~Couche~Médiane  
Caracteristiques~générales  
Changement~d' échelle  
Charger  
Choix~du~mode~d' écran  
Composition~d' images  
Concepts  
Conditions~d' utilisation  
Convolution  
Convolution~croisée  
Convolution~discrète  
Convolution~par~une~fonction~analytique  
Convolution~par~une~matrice  
Convolve  
Copyright  
Correction~pixel~par~pixel  
Corrélation  
Couche~suivante  
CurrentImageName  
dans~le~domaine~fréquentiel  
dans~le~domaine~spatial  
DeleteBuffer  
Division  
Déconvolution  
Défaire  
Définir~Masque  
Définit~composantes~RVB  
Démarrer~animation  
Echelle~de~visualisation  
Editer  
Effacer~tous~les~graphiques  
Effacer~une~image  
Eliminer~le~bruit~présent~dans~l' image  
Enregistrement  
Exemples~pratiques  
Extraire  
Faire~une~image~couleur~avec~trois~images~monochromes  
Faire~varier~contraste~et~luminosité~d' une~image  
Filtrage~non~linéaire

---

FITS  
Fonction~non~implémentée  
Fonte...  
Format~IFF~ILBM  
Gestion~des~Images  
Générer~Image  
Graphiques  
Guide~d'~utilisation  
HELP~!  
Histogramme  
Histogramme  
Historique  
IMA  
Images~monochromes  
Information~  
Informations  
Installation  
Intégrer~dans~l'~image  
Introduction  
Libère~BitMaps  
Lynx~Version~1.08  
LynxEnd  
LynxFormat  
Mask  
Mask\_Status  
Menus~de~l'~écran~de~contrôle  
Menus~de~l'~écran~de~visualisation  
Messages~d'~erreurs~ARexx  
Méthode  
Mode~'normal'  
Mode~Hold~And~Modify  
Modes  
Mult  
MultConstant  
Normalisation  
Normalize  
Objectifs  
Objet~(datatypes)  
Opérations~arithmétiques  
Ouvrir~une~image~à~l'~aide~des~datatypes  
Ouvrir~une~image~au~format~FITS  
Ouvrir~une~image~au~format~IFF  
Ouvrir~une~image~au~format~IMA  
Ouvrir~une~image~au~format~Lynx  
Ouvrir~une~image~au~format~PIC  
PIC  
Piles~d'~images  
Quitter  
Redimensionnement  
Remerciements  
Réaliser~un~'masque~flou'  
Réaliser~un~effet~du~style~'peinture~à~l'huile'.  
Réaliser~une~convolution~dans~le~domaine~de~Fourier  
Réaliser~une~déconvolution~dans~le~domaine~de~Fourier  
Réglage~des~seuils  
Rééchantillonnage  
RotateImage

---

Rotation  
Sauvegarder~une~image~au~format~FITS  
Sauvegarder~une~image~au~format~IFF  
Sauvegarder~une~image~au~format~IMA  
Sauvegarder~une~image~au~format~Lynx  
Sauver  
Sauver~Ecran  
Sauver~Fenêtre  
SelectImage  
Sigma  
Sélection~d'~une~image  
Sortir  
Soustraction  
Statistiques  
Stéphane~Poirier  
SubImage  
Théorie  
ToolTypes  
Tout~Effacer  
Traitements~d'~images  
Tramage  
Transcodage  
Transcodage~d'~image  
Transformation~de~Fourier  
Transformations~géométriques  
Transforme~RVB~vers~TSI  
Transforme~TSI~vers~RGB  
Transformée~directe  
Transformée~inverse  
Transition  
Translation  
Un~peu~de~reconnaissance~de~formes...  
Utilisation~d'~ARexx  
Visualisation  
Visualisation~d'~images~en~trichromie  
Visualisation~d'~images~monochromes  
Visualisation~en~mode~'pseudo-couleurs'  
Visualisation~en~mode~contour  
Visualisation~en~mode~mosaïque  
Vitesse~d'~animation

---