

Inserm

Institut national
de la santé et de la recherche médicale



Responsables scientifiques : Christophe CHAMOT ; Tristan PIOLOT

IMAGERIE NUMERIQUE EN BIOLOGIE

du Mardi 13 au Vendredi 16 mars 2007

à Paris - Jussieu

INTERVENANTS :

**Aude JOBART
Christophe CHAMOT
Tristan PIOLOT
Yves TOURNEUR
Fabrice CORDELIÈRES**

Inserm



Institut national
de la santé et de la recherche médicale



Christophe CHAMOT

Plateforme de recherche

« Imagerie des processus dynamiques en Biologie Cellulaire et Biologie du Développement »

UMR7592 – Institut Jacques Monod – Paris

Tél. : 01 44 27 57 84

chamot@ijm.jussieu.fr

Fabrice CORDELIERES

Institut Curie

Section de recherche/CNRS UMR 146

Plateforme d'Imagerie Cellulaire

Bâtiment 112 – Centre Universitaire

91405 Orsay Cedex

Tel : 01 69 86 31 30

Fabrice.cordelieres@curie.u-psud.fr

Aude JOBART

Plateforme de recherche

« Imagerie des processus dynamiques en Biologie Cellulaire et Biologie du Développement »

UMR7592 – Institut Jacques Monod – Paris

Tél. : 01 44 27 57 84

jobart@ijm.jussieu.fr

Tristan PIOLOT

Plateforme de Recherche

"Imagerie des processus dynamiques en Biologie Cellulaire et Biologie du Développement"

UMR7592-Institut Jacques Monod, CNRS, Université Paris 6 et 7

2, place Jussieu – Tour 43

75251 PARIS cedex 05

Tel : 01 44 27 57 84

piolot@ijm.jussieu.fr

Yves TOURNEUR

Centre Commun de Quantimétrie

Université Lyon1

8 Avenue de Rockefeller

69373 Lyon Cedex 08

Tél. : 04 78 77 75 19

<http://quantimetrie.univ-lyon1.fr>

Imagerie Numérique

Programme

Mardi 13 Mars :

- 9h00-9h30 Accueil
- 9h30-12h00 Formation et Acquisition d'une image numérique *A. JOBART-MALFAIT*
- 14h00-14h45 Stockage et édition numérique (formats de fichiers) *C.CHAMOT*
- 14h45-15h45 Introduction à ImageJ *C. CHAMOT ;*
- 15h45-16h00 Pause
- 16h00-17h30 T.P. ImageJ . Prise en main et opérations courantes *C. CHAMOT ; A. JOBART*

Mercredi 14 Mars :

- 9h30-10h30 Filtrage spatial *T. PIOLOT ; C. CHAMOT*
- 10h30-11h30 T.P. Filtrage spatial *T. PIOLOT ; C. CHAMOT ; Y. TOURNEUR*
- 11h30-12h15 Filtrage fréquentiel *Y. TOURNEUR*
- 12h15-13h00 Introduction aux ondelettes *T. PIOLOT ; C. CHAMOT*
- 14h00-15h00 Applications du filtrage fréquentiel et des ondelettes *Y.TOURNEUR, T. PIOLOT ; C. CHAMOT*
- 15h00-15h30 Morpho-mathématique. *T. PIOLOT ; C. CHAMOT*
- 15h30-15h45 Pause
- 15h45-16h45 T.P. morpho-mathématique *T. PIOLOT ; C. CHAMOT ; Y. TOURNEUR*
- 16h45-17h30 Exemple d'applications *C. CHAMOT ; T. PIOLOT*

Jeudi 15 Mars :

- 9h30-12h30 Co-localisation et T.P. *F. CORDELIÈRES, S. BOLTE*
- 14h00-17h00 Déconvolution et Représentation 3D *C.CHAMOT, T. PIOLOT*

Vendredi 16 Mars :

Introduction à la programmation en imagerie

- 9h30-12h00 Macros sous ImageJ (automatisations de tâches, traitement récursif) *Cours : T. PIOLOT*
- 14h00-16h00 Macros & plugins : Applications (ex. de tracking) *C. CHAMOT ; T. PIOLOT ; A. JOBART-MALFAIT*
- 16h00-17h00 Évaluation
- Pot de fin de stage

Aude JOBART

Formation de l'image

Formation et acquisition d'une image numérique

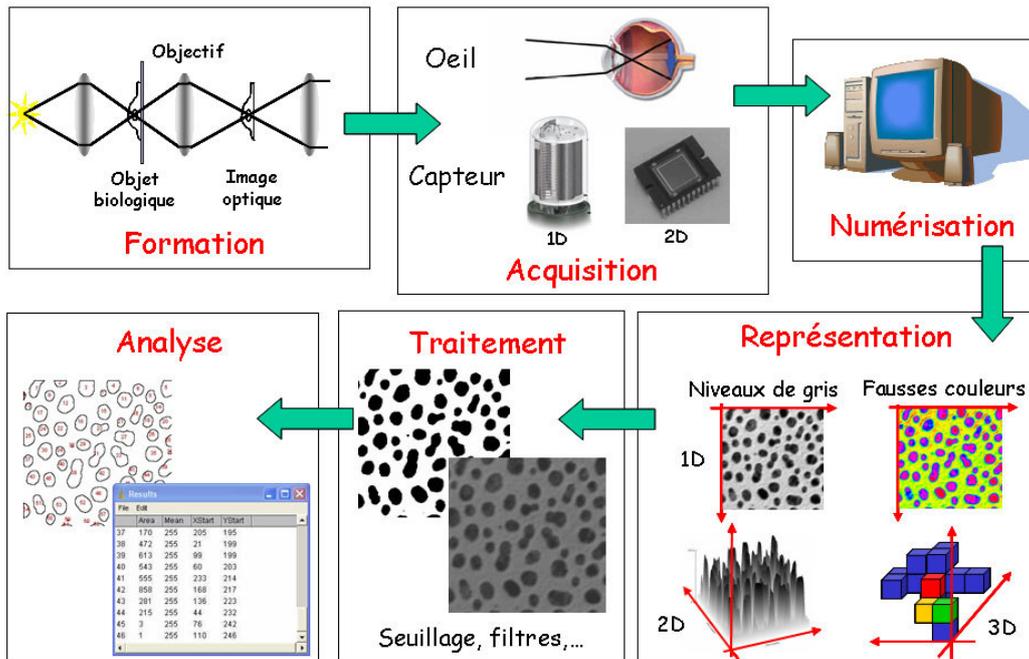
1	La chaîne de mesure de l'image	3
2	Formation de l'image	3
2.1	<i>Généralités sur la lumière</i>	3
2.1.1	Nature de la lumière	3
2.1.2	Aspect corpusculaire	4
2.1.3	Aspect ondulatoire	4
2.2	<i>Approche géométrique de la formation de l'image</i>	5
2.2.1	Rappels d'optique	5
2.2.2	Réfraction et réflexion	5
2.3	<i>Approche ondulatoire de la formation de l'image</i>	6
2.3.1	Interférences	6
2.3.2	Diffraction	7
2.3.3	Diffusion	7
2.3.4	Absorption	7
3	Microscopie en lumière blanche	8
3.1	<i>Objets de phase et d'amplitude</i>	8
3.2	<i>Le microscope optique</i>	8
3.2.1	L'objectif	8
3.2.2	Trajets des rayons lumineux	10
3.2.3	Génération de contraste en lumière blanche	10
4	Microscopie de fluorescence	11
4.1	<i>Le phénomène de fluorescence</i>	11
4.2	<i>Les caractéristiques de la fluorescence</i>	12
4.2.1	Rendement quantique de fluorescence	12
4.2.2	Coefficient d'extinction	12
4.2.3	Durée de vie de fluorescence	12
4.3	<i>Le microscope à fluorescence</i>	12
5	Résolution optique	12
5.1	<i>La Point Spread Function</i>	12
5.2	<i>La fonction de transfert de contraste</i>	13
5.3	<i>Résolution latérale et axiale</i>	13
5.3.1	Résolution latérale	14
5.3.2	Résolution axiale	14
6	Acquisition de l'image	14
6.1	<i>Les différents types de capteurs</i>	14
6.1.1	Structure des détecteurs point par point	14
6.1.2	Structure des capteurs CCD	15
6.2	<i>Les différents types de capteurs CCD</i>	15
6.2.1	Full frame	15
6.2.2	Frame Transfert	15
6.2.3	Interligne Transfert	16
6.3	<i>Caractéristiques des capteurs CCD</i>	16
6.3.1	Sensibilité spectrale	16
6.3.2	Rapport signal/bruit	16
6.3.3	Résolution spatiale	17

6.3.4	Dynamique de mesure et de numérisation.....	17
6.4	<i>Les caméras à amplification de signal.....</i>	<i>17</i>
6.4.1	La technologie EMCCD	17
6.4.2	Les CCD intensifiées	18
6.5	<i>Les caméras CCD couleurs.....</i>	<i>18</i>
6.5.1	Les CCD à matrice de Bayer	18
6.5.2	Les CCD monochromes avec roue de filtres ou filtre de cristaux liquides	19
6.5.3	Les tri-CCD	19
6.5.4	Les CMOS 3 couches.....	19
7	Dégradation de l'image.....	20
7.1	<i>Sources de dégradation.....</i>	<i>20</i>
7.1.1	Au niveau de l'objet	20
7.1.2	Au niveau du microscope	20
7.1.3	Au niveau du capteur.....	20
7.2	<i>Correction des dégradations.....</i>	<i>20</i>

1 La chaîne de mesure de l'image

L'image d'un objet est une représentation de la répartition spatiale d'une grandeur physique de cet objet. L'image est dite en 2D si les variations de la grandeur physique se produisent dans 2 dimensions de l'espace (XY), et 3D si ces variations se produisent dans les 3 dimensions de l'espace (XYZ). Chaque élément de l'image (ou pixel, pour Picture Element) représente un élément de volume (ou voxel, pour Volume PiXture Element) de l'échantillon exploré.

En microscopie optique, une image est obtenue grâce à l'interaction de la lumière visible avec l'objet observé. D'autres types de rayonnement peuvent être utilisés pour obtenir des images : électrons en microscopie électronique, rayons X en radiologie, ultra-sons en échographie...



L'image agrandie de l'objet est formée par l'interaction de rayons lumineux dans le microscope optique constitué de plusieurs lentilles. L'image est formée dans le plan image, plan au niveau duquel on place un montage optique permettant de former une image rétinienne dans l'œil. C'est également au niveau de ce plan image que l'on place le capteur. En microscopie, le capteur a pour fonction de transformer les variations d'intensité lumineuses en un signal électrique. Après numérisation du signal électrique, l'image est un tableau de valeurs d'intensités lumineuses. On peut utiliser différents types de représentation du signal ainsi que différents codes couleurs. Le signal peut ensuite être traité mathématiquement et analysé pour en tirer des informations quantitatives.

2 Formation de l'image

2.1 Généralités sur la lumière

2.1.1 Nature de la lumière

La question de la nature de la lumière a fait l'objet de vives discussions pendant des siècles. Deux théories s'opposent : la théorie corpusculaire et la théorie ondulatoire.

La lumière doit être décrite en utilisant simultanément les concepts d'onde et de particule (i.e dualité onde-particule). La lumière est à la fois un phénomène ondulatoire, d'où la notion de longueur d'onde et un phénomène corpusculaire, comme en témoigne l'existence des photons.

La dualité onde-particule de la lumière relie l'énergie d'un photon (théorie corpusculaire) à la fréquence de l'onde électromagnétique (théorie ondulatoire) :

$$E = h \cdot \nu$$

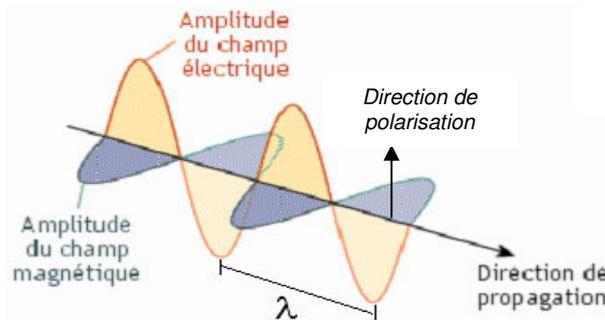
où E est l'énergie d'un photon en joules, h est la constante de Planck ($h = 6,62 \times 10^{-34}$ J·s) et ν est la fréquence de la lumière en hertz.

2.1.2 Aspect corpusculaire

La théorie corpusculaire développée par Newton et Planck décrit la lumière comme composée de particules, les photons. Ces photons ont une masse nulle et sont dépourvus de charge électrique. La petite quantité d'énergie transportée par un photon est dépendante de la longueur d'onde qui lui est associée. Cette énergie est d'autant plus grande que la longueur d'onde est courte.

2.1.3 Aspect ondulatoire

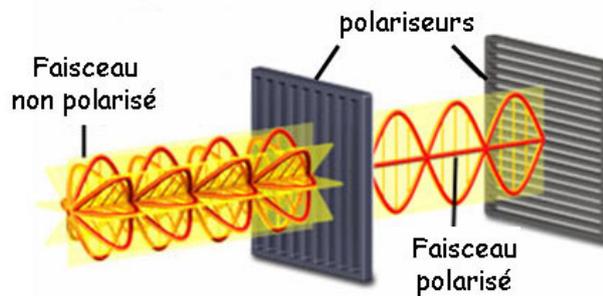
Une onde lumineuse est constituée de la superposition d'un champ électrique E et d'un champ magnétique B, oscillant de manière sinusoïdale dans une même direction de propagation. Pour une approche simplifiée on considère généralement uniquement la variation de l'amplitude du champ électrique associé à l'onde.



Représentation schématique d'une onde électromagnétique

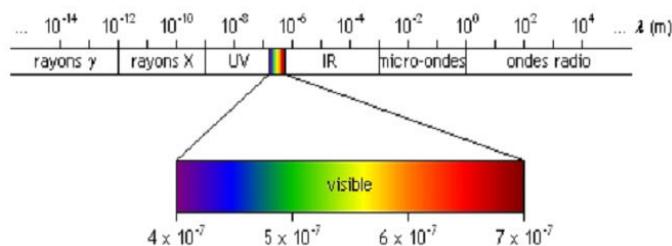
Lorsque l'orientation du plan de vibration est aléatoire par rapport au plan du champ électrique E, la lumière est dite non polarisée (source lumineuse « classique »).

Lorsque le plan de vibration est orienté de la même façon que le plan du champ électrique, la lumière est polarisée (cas des lasers).



Polarisation de la lumière

Le spectre électromagnétique couvre l'étendue complète des ondes électromagnétiques de toutes longueurs d'onde. La lumière visible ne constitue qu'une petite partie du spectre électromagnétique. Des petites longueurs d'onde aux grandes longueurs d'onde on retrouve les rayons gamma, les rayons X, les ultraviolets (UV), la lumière visible (de 400 nm (violet) à 700 nm (rouge)), les infrarouges (IR), les micro-ondes et les ondes radio.



Spectre électromagnétique

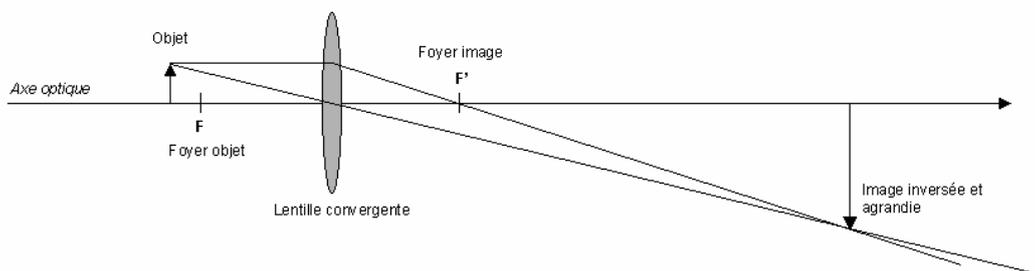
Une onde électromagnétique est caractérisée par sa longueur d'onde λ en mètre (trajet parcouru par l'onde pendant une période), et sa fréquence ν en hertz (nombre de perturbations en 1 seconde). Ces deux paramètres sont reliés par la formule : $\lambda=c/\nu$ où c est la vitesse de la lumière dans le vide (300.000 km.s⁻¹).

2.2 Approche géométrique de la formation de l'image

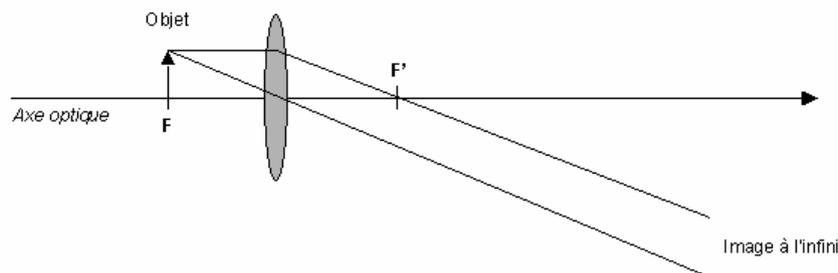
En optique géométrique, la lumière est vue sous sa forme corpusculaire uniquement : un photon se déplace en ligne droite dans le vide et dans un milieu homogène. Sa trajectoire crée un rayon lumineux.

2.2.1 Rappels d'optique

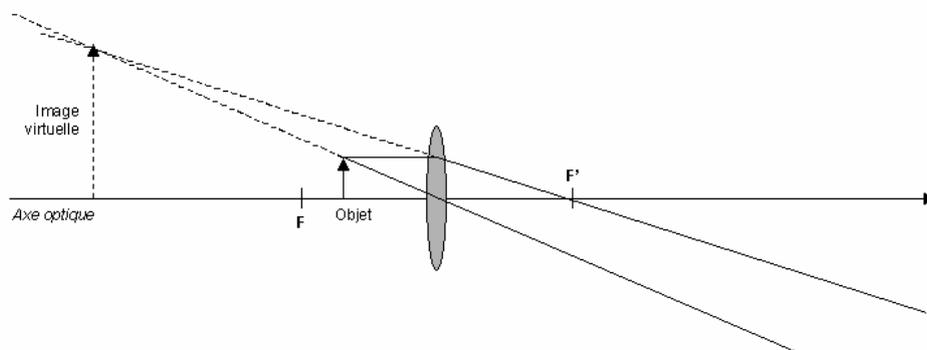
Les lentilles présentes dans les objectifs sont des lentilles convergentes. L'objet à étudier est placé en amont du foyer objet (F) de la lentille. L'image formée par la lentille se forme au delà de son foyer image (F'). Le grandissement dépend de (i) la distance du foyer au centre de la lentille et (ii) la distance de l'objet au foyer objet (F). Plus la distance objet-foyer objet sera courte, plus fort sera le grossissement.



Si l'objet se trouve au foyer objet (F) de la lentille, les rayons qui émergent de la lentille sont parallèles. L'image de l'objet se formera à l'infini :



Si l'objet se trouve entre le foyer objet et la lentille, aucune image ne se forme après la lentille. L'image créée est dite virtuelle, elle est agrandie et se forme avant l'objet :



2.2.2 Réfraction et réflexion

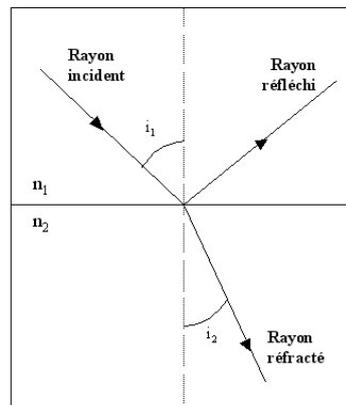
Un milieu translucide qui permet le passage de la lumière possède un indice de réfraction noté n . Cet indice dépend des propriétés intrinsèques du matériau. Il correspond à une mesure de sa densité optique.

$$n=c/\nu$$

où c est la vitesse de la lumière dans le vide et v est la célérité de la lumière dans le milieu traversé.

Lorsqu'un rayon rencontre un milieu d'indice différent du sien, il est soumis aux lois de la réfraction. Cette réfraction est décrite par la loi de Snell-Descartes :

$$n_1 \cdot \sin(i_1) = n_2 \cdot \sin(i_2)$$



Réfraction d'un rayon lumineux

La surface séparant les deux milieux est appelée dioptre. On appelle angle d'incidence (i_1) l'angle entre le rayon incident et la normale au dioptre, et angle de réfraction (i_2) l'angle entre le rayon réfracté et la normale.

Si l'angle d'incidence est nul (rayon perpendiculaire au dioptre), il n'y a pas de déviation.

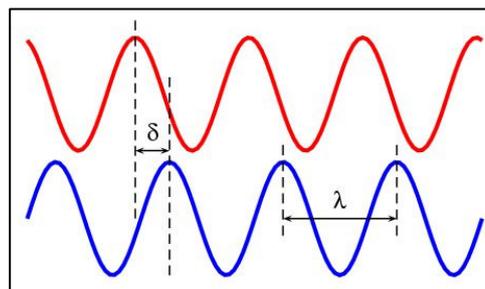
Si on dépasse un angle d'incidence limite, le rayon n'est plus transmis : il y a réflexion totale.

2.3 Approche ondulatoire de la formation de l'image

2.3.1 Interférences

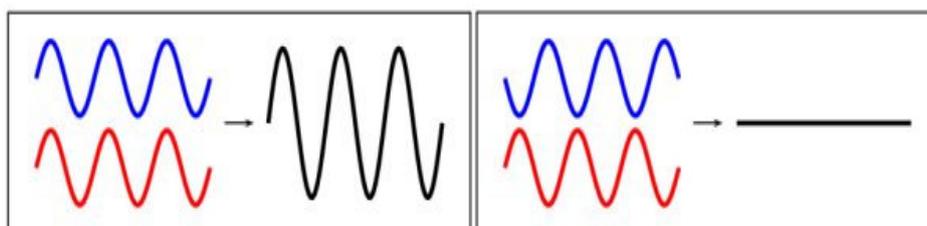
Les ondes étant des grandeurs vectorielles, deux d'entre elles, ou davantage, peuvent se composer et s'ajouter vectoriellement en un point de l'espace où elles se situent au même instant.

Lorsque deux ondes lumineuses, en provenance de la même source, atteignent le même point de l'espace en ayant parcouru des chemins différents pour y parvenir, elles arrivent déphasées ; le schéma ci-dessous montre le décalage ou différence de marche δ entre ces deux ondes, ainsi que leur longueur d'onde commune λ .



Lorsque les ondes ont parcouru exactement le même chemin, l'intensité obtenue est maximale puisque les deux ondes s'ajoutent en phase. Il en va d'ailleurs de même à chaque fois que la différence de marche δ entre ces deux ondes est égale à la longueur d'onde λ , ou à un nombre entier de longueurs d'onde. On parle, dans ce cas, d'interférences constructives.

Par contre, à chaque fois que la différence de marche sera égale à une demi-longueur d'onde, ou à un nombre impair de demi-longueurs d'onde, les ondes arrivent en opposition de phase et l'intensité est nulle.



Interférences constructives et destructives

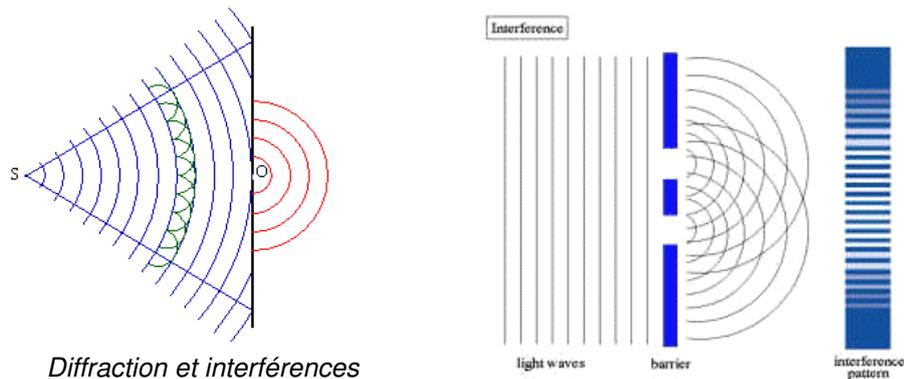
2.3.2 Diffraction

Les ondes sont diffractées lorsqu'elles rencontrent un obstacle ou une ouverture de dimension voisine de leur longueur d'onde. Selon la forme de l'ouverture (trou ou fente), le pattern de diffraction est différent.

Le phénomène de diffraction est décrit par le principe de Huygens-Fresnel :

Supposons une onde sphérique émise par un point lumineux S abordant un écran percé d'un trou O. L'expérience montre que l'onde n'est pas transmise sous sa forme initiale, mais que tout se passe comme si le trou O émettait à son tour une onde sphérique. Cette ouverture se comporte comme une source secondaire de même fréquence que la source primaire créant l'onde. On dit que le point O diffracte la lumière.

Ce principe est généralisable et l'on peut considérer chaque point d'une surface d'onde comme un point source émettant une ondelette sphérique.



Si un écran percé de plusieurs trous est éclairé par une onde plane, les ondelettes produites par les sources créées peuvent interférer entre elles. Les amplitudes des ondes sphériques s'ajoutent, créant dans certaines zones des interférences positives et dans d'autres des interférences négatives. Le pattern de diffraction représente donc une alternance de zones lumineuses et sombres.

Dans un microscope, les rayons lumineux vont interférer entre eux, ce qui aura pour conséquence d'altérer le contraste et la résolution des images formées. Cependant, ces « défauts » peuvent être mis à profit pour créer des contrastes (microscopie à contraste interférentiel).

2.3.3 Diffusion

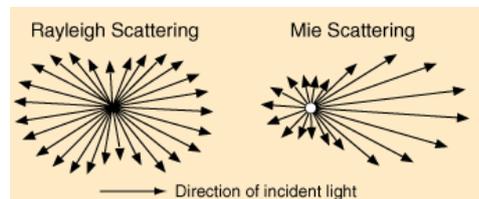
La diffusion est le phénomène par lequel la lumière est déviée dans de multiples directions. Elle peut avoir lieu à la rencontre d'une interface entre deux milieux ou à la traversée d'un milieu.

On distingue différents types de diffusion selon la taille relative des cibles par rapport à la longueur d'onde de la radiation incidente.

Le processus de diffusion est le plus souvent « élastique », c'est-à-dire qu'il a lieu sans changement de fréquence ou de longueur d'onde des rayonnements composant le faisceau (diffusion de Rayleigh ou de Mie).

La diffusion Rayleigh est la diffusion par des particules d'une taille d'une dizaine de nanomètres.

La diffusion de Mie s'observe lorsque la taille des particules diffusantes est plus importante (entre 0.1 et 10 fois la longueur d'onde).



Lorsque la diffusion engendre un changement de longueur d'onde et de fréquence, elle est dite « inélastique » (diffusion Raman).

2.3.4 Absorption

Lorsqu'un rayon lumineux traverse un matériau, les trains d'ondes vont interagir avec la matière. Certains photons vont être absorbés par les molécules du matériau. En conséquence, l'intensité I mesurée en sortie du matériau sera diminuée par rapport à l'intensité I_0 mesurée avant l'entrée dans le matériau. Le rapport I_0/I constitue l'absorbance du milieu.

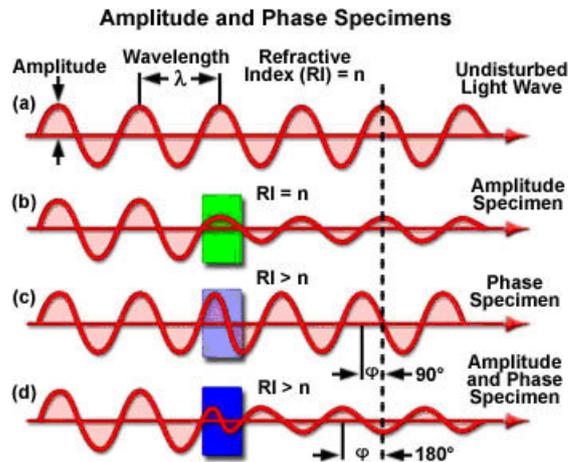
Pour certains matériaux l'absorbance peut varier en fonction de la longueur d'onde. Cette propriété permet de filtrer la lumière afin de ne retenir qu'un certain contingent de longueur d'onde.

3 Microscopie en lumière blanche

3.1 Objets de phase et d'amplitude

Pour les objets transparents, on distingue 2 types d'objets :

- Les objets de phase qui modifient la phase des ondes qui les traversent mais pas leur amplitude. Ces changements de phase sont dus majoritairement aux différences d'indice de réfraction entre l'objet et le milieu environnant.
- Les objets d'amplitude réduisent l'amplitude de la lumière et donc son intensité.



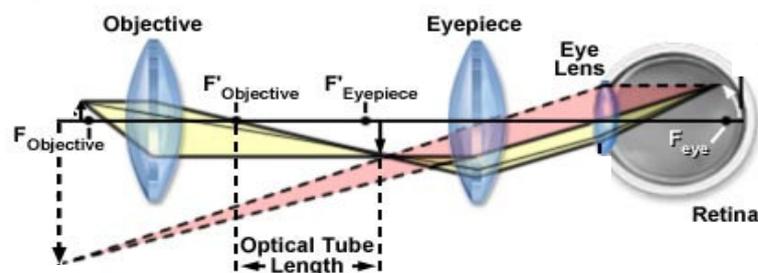
3.2 Le microscope optique

Le microscope est un montage optique qui permet d'obtenir une image agrandie d'un objet. Il doit donc :

- produire une image agrandie de l'échantillon,
- séparer les détails
- rendre ces détails visibles par l'œil ou un capteur.

Le montage le plus simple est constitué d'une source lumineuse (qui illumine l'objet), d'un objectif (formant l'image) et d'oculaires qui permettent de voir l'image à l'œil.

Ce type de montage est appelé microscope plein champ.



3.2.1 L'objectif

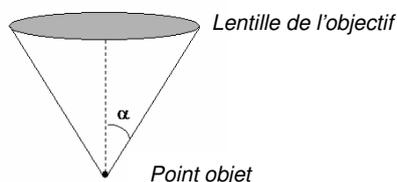
Le constituant majeur d'un microscope est l'objectif. C'est de lui que dépend le grossissement et la résolution d'un microscope.

3.2.1.1 Ouverture numérique

On définit l'angle α comme le demi angle du cône de lumière pouvant être récupéré par l'objectif et formé par les rayons issus d'un point objet. L'ouverture numérique (O.N) de l'objectif est fonction de cet angle α et de l'indice de réfraction du milieu n . Elle s'exprime de la façon suivante :

$$\text{O.N.} = n \cdot \sin \alpha$$

α : demi-angle d'ouverture de l'objectif
 n : indice de réfraction du milieu

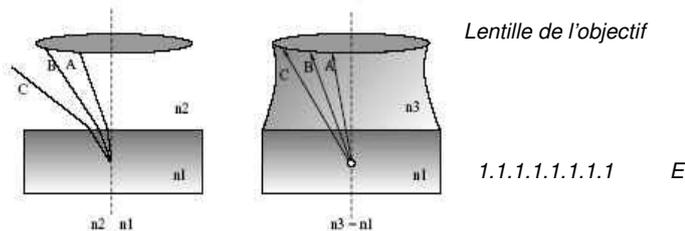


L'ouverture numérique est une caractéristique essentielle de l'objectif. C'est elle qui est directement responsable de sa luminosité et de sa résolution.

Deux autres caractéristiques de l'objectif dépendent de l'ouverture numérique : la distance de travail et la profondeur de champ. Un objectif de grande qualité sera caractérisé par une grande ouverture numérique, ce qui a pour conséquence une distance de travail limitée et une profondeur de champ réduite.

3.2.1.2 Immersion

Afin de limiter l'angle de réfraction à la sortie de l'échantillon, on utilise des objectifs à immersion. Le milieu utilisé pour l'immersion est de plus souvent de l'huile, de l'eau ou du glycérol. Les différences d'indice entre les milieux traversés seront moins importantes, l'objectif pourra ainsi récolter plus de rayons lumineux.



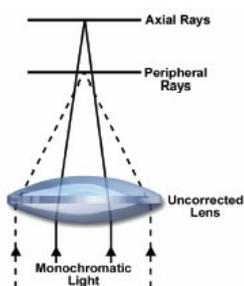
A gauche : la différence entre les indices de réfraction n_1 et n_2 est grande. Les rayons lumineux sont déviés vers l'extérieur lors du passage du milieu de montage à l'air. Les rayons les plus externes sont donc perdus.

A droite : on utilise un liquide d'immersion d'indice n_3 proche de n_1 . Les rayons ne sont plus déviés et tous peuvent être collectés par la lentille.

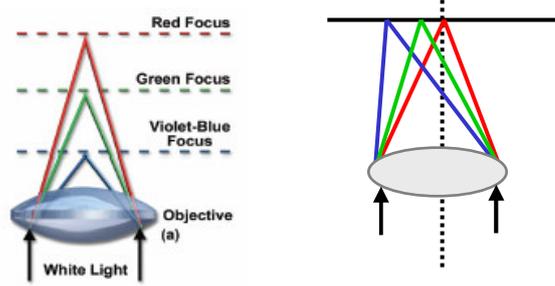
3.2.1.3 Aberrations

Les images formées par un système optique, même seulement constitué d'une seule lentille comportent des défauts. On en distingue trois types :

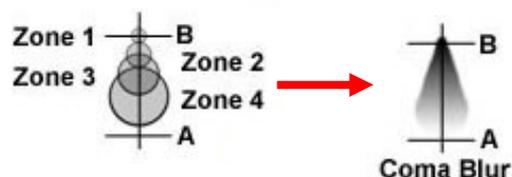
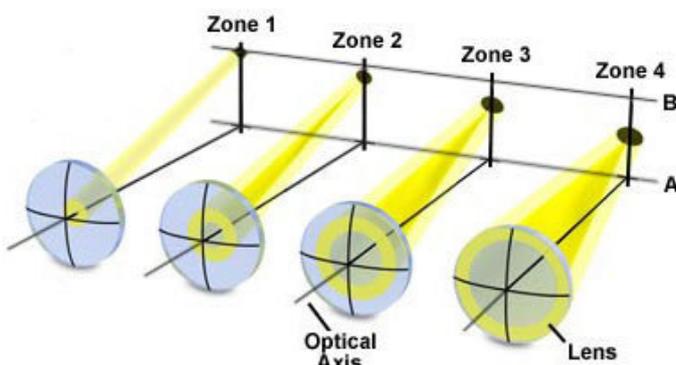
- Les aberrations chromatiques : lorsqu'un rayon lumineux composé de plusieurs longueurs d'ondes traverse une lentille convexe, les couleurs ne sont pas focalisées au même endroit.
- Les aberrations de sphéricité : les rayons qui passent au centre de la lentille ne focalisent pas au même endroit (plus loin) que ceux passant vers les extrémités de la lentille.
- La coma : c'est une aberration due à une différence de focalisation des rayons ayant un angle important par rapport à l'axe optique. Les rayons arrivant au centre de la lentille et ceux arrivant en périphérie focalisent dans le même plan mais pas au même endroit. Le résultat est une image déformée rappelant la queue d'une comète.
- L'aberration de planéité de champ : la mise au point ne peut être faite simultanément sur les bords et au centre du champ. Cela se traduit par une image à l'aspect "bombé".



Aberration de sphéricité



Aberration chromatique axiale et longitudinale



Aberration de coma

Les objectifs peuvent être corrigés pour ces aberrations. On distingue :

- les objectifs achromat corrigés pour les aberrations chromatiques dans 2 longueurs d'onde, et pour les aberrations sphériques dans une seule longueur d'onde.
- les objectifs apochromat corrigés pour les aberrations chromatiques dans 3-4 longueurs d'onde, et pour les aberrations sphériques dans 4-5 longueurs d'onde.
- les objectifs plan corrigés pour la planéité du champ.
- certains objectifs sont corrigés à la fois pour des aberrations chromatiques, sphériques et pour la planéité du champ. Ce sont les objectifs dits plan-apo.

3.2.2 Trajets des rayons lumineux

Le montage le plus couramment utilisé pour optimiser l'illumination (résolution et contraste) d'un objet est l'**alignement de Köhler**.

Dans ce montage, deux séries de plans conjugués images sont indépendantes.

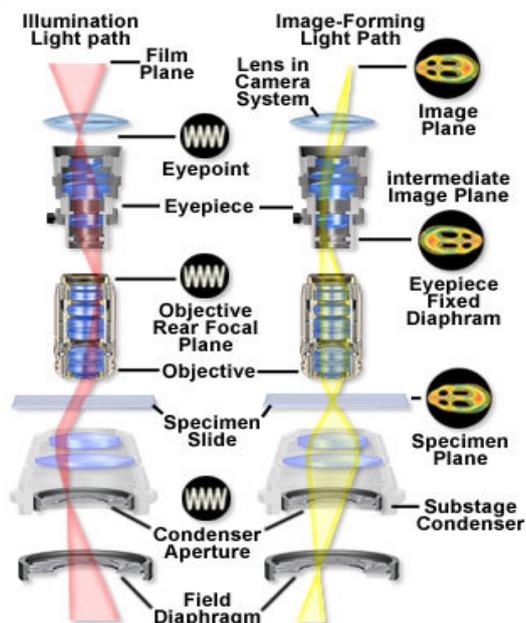
- Une lentille collectrice placée en aval de la source lumineuse produit une image de la source au niveau du diaphragme d'ouverture. Le diaphragme étant situé au niveau du plan focal objet du condenseur, c'est un faisceau de rayons parallèles qui vient illuminer l'échantillon et qui focalisent sur le plan focal arrière de l'objectif puis sur la lentille des oculaires.

L'ouverture et la fermeture du diaphragme d'ouverture contrôlent l'angle du cône de lumière qui atteint l'échantillon.

- Un autre diaphragme, le diaphragme de champ, est placé en amont du condenseur. Son rôle est de contrôler le diamètre du faisceau lumineux atteignant l'échantillon. L'image de ce diaphragme se forme dans le plan de l'échantillon. C'est un faisceau de rayons parallèles qui traverse donc l'objectif.

Il est possible, à ce niveau, de contrôler la résolution optique de l'image via l'ouverture ou la fermeture du diaphragme d'ouverture de l'objectif.

Séries de plans conjugués dans l'illumination de Köhler



La microscopie en lumière transmise n'offre pas beaucoup de contraste pour les objets transparents comme les cellules. D'autres techniques permettent d'obtenir un meilleur contraste : la microscopie à contraste de phase et la microscopie à contraste interférentiel. Ces deux types de microscopie permettent de transformer une différence d'indice de réfraction, à laquelle l'œil n'est pas sensible, en une différence d'intensité.

3.2.3 Génération de contraste en lumière blanche

3.2.3.1 Microscope à contraste de phase

Le principe repose sur la formation d'un contraste d'intensité créé par l'interférence de rayons lumineux. Des anneaux de phase sont insérés dans le trajet optique. Ils vont créer des retards de phase différents suivant que les rayons traversent ou non l'objet. Le déphasage entre ces rayons créé le contraste et permet de mettre en évidence des structures dans l'objet observé.

3.2.3.2 Microscope à contraste interférentiel (DIC ou Nomarski)

Le principe de ce type de microscopie repose sur la division d'un rayon polarisé en 2 rayons de même longueur d'onde mais polarisés perpendiculairement par un prisme de Wollaston. Ces deux rayons traversent l'objet en des points séparés mais très proches. Suivant le milieu traversé (noyau, cytoplasme, ...), les rayons subissent un décalage de phase différent. Suivant la différence de phase entre les rayons, un contraste positif ou négatif sera créé, permettant là encore de mettre en évidence des structures cellulaires.

4 Microscopie de fluorescence

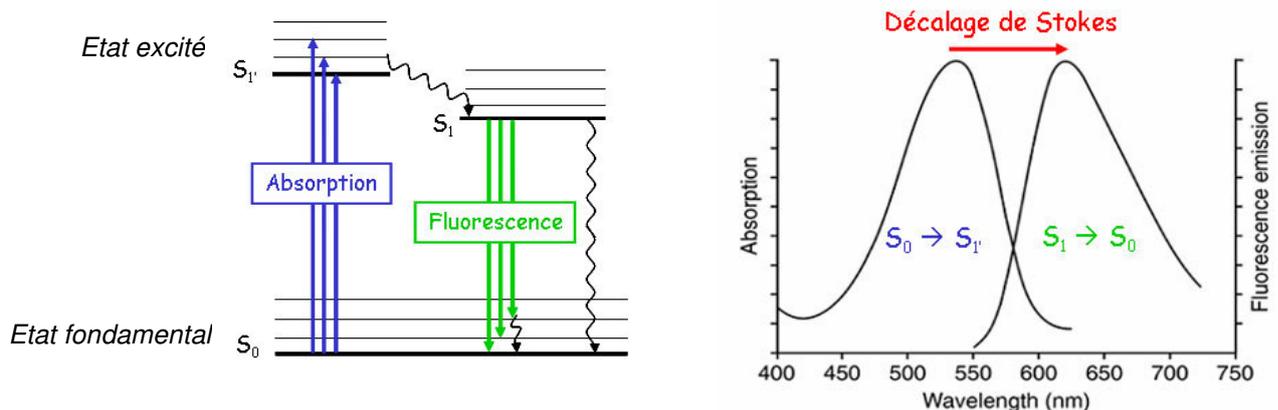
4.1 Le phénomène de fluorescence

Une molécule ou un atome peut absorber de l'énergie à partir de sources diverses : rayonnement électromagnétique, réarrangements électroniques, réaction chimique exothermique... Elle est alors dans un état excité instable et son retour à l'état fondamental peut se produire de différentes façons : luminescence, transfert d'énergie intra ou intermoléculaire, quenching, ...

La luminescence résulte du retour d'une molécule ou d'un atome excité vers un état fondamental par l'émission d'un photon. Il existe de nombreux types de luminescence qui se distinguent par la source d'activation de la molécule (thermo, photo, électro, sono, bioluminescence)

La fluorescence est un cas particulier de photoluminescence. La source d'excitation de la molécule est un rayonnement électromagnétique, la lumière.

Le phénomène de fluorescence peut se résumer en 3 étapes, que l'on peut schématiser de la façon suivante (diagramme de Jablonski) :



Etape 1

Un photon d'énergie $h\nu_{EX}$, fourni par une source lumineuse externe telle qu'un laser ou une lampe à mercure, est absorbé par la molécule, permettant le passage d'un état énergétique fondamental S_0 à un état excité S_1' , d'énergie supérieure à S_0 .

Etape 2

Cet état excité dure généralement un temps très limité, entre 10^{-8} et 10^{-10} s. Pendant ce temps la molécule subit des changements conformationnels et elle est soumise à de multiples interactions avec son environnement. L'énergie de S_1' est alors partiellement dissipée. La molécule possède alors une énergie interne S_1 plus faible que l'énergie S_1' initiale.

Etape 3

Le retour à l'état fondamental S_0 se fait par émission d'un photon $h\nu_{EM}$. Du fait de la dissipation d'énergie pendant l'état excité, l'énergie du photon ré-émis est plus faible que celle du photon absorbé. La longueur d'onde du photon émis est donc supérieure à celle du photon absorbé..

Chaque molécule peut être caractérisée par des spectres d'absorption et d'émission qui lui sont propres et qui reflètent la distribution de probabilité des transitions énergétiques.

Généralement, le spectre d'émission de fluorescence est une image inversée (effet miroir) du spectre d'absorption. L'écart de longueur d'onde entre les deux maxima est appelé décalage de Stokes.

4.2 Les caractéristiques de la fluorescence

4.2.1 Rendement quantique de fluorescence

Le rendement quantique de fluorescence correspond au rapport du nombre de photons émis sur le nombre de photons absorbés par une molécule.

Lorsqu'un photon est absorbé, la molécule excitée peut revenir à son état fondamental par des processus radiatifs (émission d'un photon) ou non radiatifs (contact du fluorophore avec une molécule extinctrice : « quenching », transfert d'énergie vers une molécule accepteuse, ...).

Le rendement quantique rend donc compte de la compétition entre les phénomènes de désexcitation radiatifs et non radiatifs. Il caractérise l'efficacité du processus de fluorescence et donc la fraction de molécules qui retournent à l'état fondamental en émettant un photon.

4.2.2 Coefficient d'extinction

L'intensité d'absorption ou coefficient d'extinction ϵ reflète la probabilité d'absorption. Plus ϵ sera élevé, plus élevée sera la fluorescence à intensité lumineuse incidente et rendement quantique égaux.

4.2.3 Durée de vie de fluorescence

La durée de vie de fluorescence se définit comme le temps moyen qu'une molécule reste à l'état excité.

Dans le cas d'une population de molécules, la durée de vie de fluorescence correspond au temps moyen de décroissance de la population de molécules excitées simultanément.

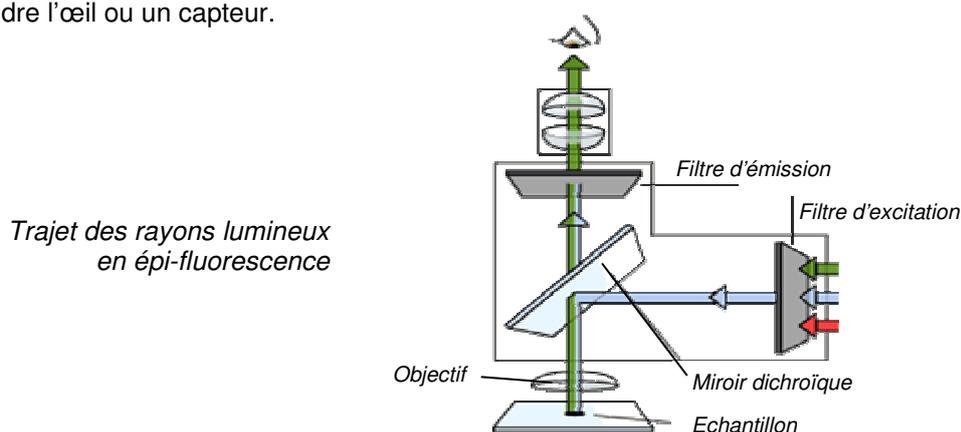
La durée de vie de fluorescence est de l'ordre de la nanoseconde.

4.3 Le microscope à fluorescence

En fluorescence, on doit pouvoir exciter l'échantillon avec une longueur d'onde donnée, et récupérer séparément la fluorescence émise à une longueur d'onde supérieure (décalée vers le rouge).

Le montage optique le plus simple pour cela est de travailler en épi-illumination. On parle alors de microscope à épi-fluorescence.

La lumière d'excitation est filtrée par un filtre d'excitation. Une bande de longueur d'onde est ainsi sélectionnée. Un miroir dichroïque réfléchit spécifiquement cette bande de longueur d'onde pour l'envoyer vers l'échantillon. La lumière d'excitation est focalisée sur l'échantillon par l'objectif. Le même objectif permet de collecter la lumière émise par l'échantillon, dans une gamme de longueur d'onde supérieure. Cette émission de l'échantillon se fait dans toutes les directions, mais seuls les rayons captés par l'objectifs traversent le miroir dichroïque. Cette lumière émise est alors filtrée par un filtre d'émission avant d'atteindre l'œil ou un capteur.



5 Résolution optique

5.1 La Point Spread Function

Comme nous avons pu le voir jusqu'à maintenant, la formation de l'image peut être décrite grâce à l'optique géométrique. Cependant, elle fait également intervenir les propriétés ondulatoires de la lumière, en particulier la diffraction.

En effet, la formation de l'image dans un microscope est également la résultante de deux fonctions de répartition de la lumière : une fonction d'étalement du point dans le plan focal conditionnant la résolution latérale (en XY) et une fonction de défocalisation conditionnant la résolution axiale (en Z).

La fonction d'étalement du point, ou PSF (Point Spread Function) est la description mathématique de l'image du disque d'Airy obtenue lorsqu'on observe un point lumineux au travers d'un objectif.

Ce disque d'Airy résulte de la formation d'interférences des ondes lumineuses diffractées le long du trajet des rayons au niveau du plan image. Les interférences constructives et destructives créent des zones lumineuses et sombres en alternance. L'image d'une source ponctuelle au travers d'un objectif est donc une tâche lumineuse entourée d'anneaux concentriques d'intensité décroissante. De la même façon, il existe une figure de diffraction le long de l'axe Z.

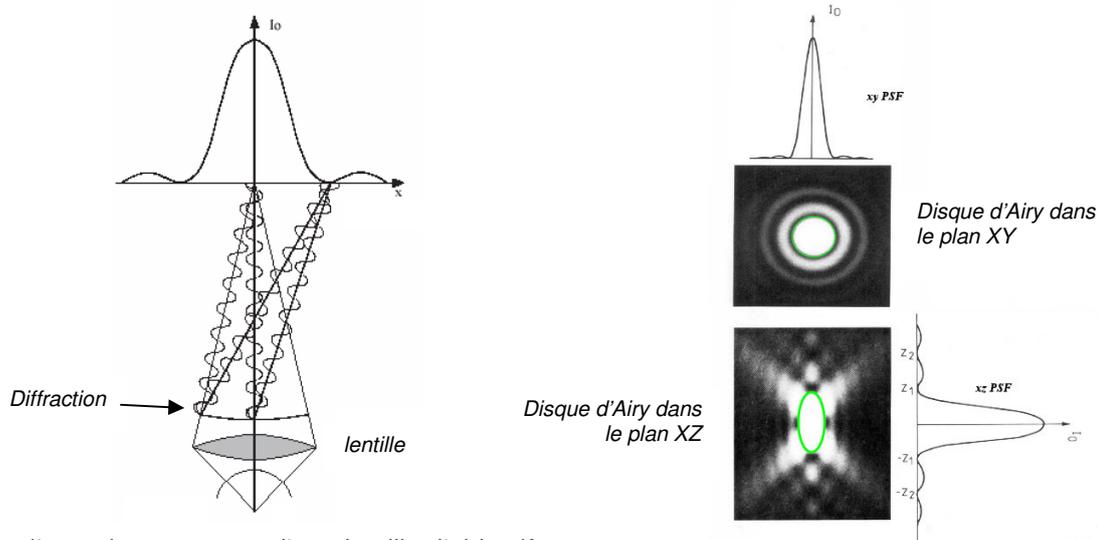


Image d'un point au travers d'une lentille d'objectif

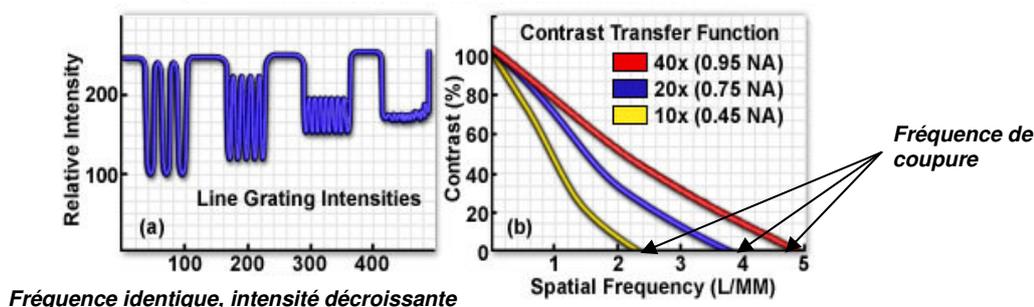
Disque d'Airy et distribution en intensité

L'image d'un objet en trois dimensions sera donc la somme des images de tous les points qui le compose. Dans un plan image, on trouvera la partie centrale de la figure de diffraction des points focalisés dans ce plan, mais également les parties supérieures et inférieures des figures de diffraction des points lumineux situés dans les plans supérieurs et inférieurs. L'image sera donc flouée et le contraste diminué.

5.2 La fonction de transfert de contraste

La résolution optique peut également être décrite par la fonction de transfert de contraste. Elle se traduit par sa capacité de restitution des fréquences spatiales. La fonction de transfert de contraste est la transformée de Fourier de la PSF. Si on mesure la variation des profils d'intensité d'un disque d'Airy, on observe que la différence d'intensité entre les pics maximum et le point d'intensité minimale va décroître graduellement.

Si on mesure la variation d'intensité de cibles test composées d'une alternance de lignes lumineuses et sombres de plus en plus rapprochées, on obtient la courbe ci-dessous. La fréquence de coupure se définit comme la fréquence à laquelle l'œil ne peut plus distinguer les différences de contraste.



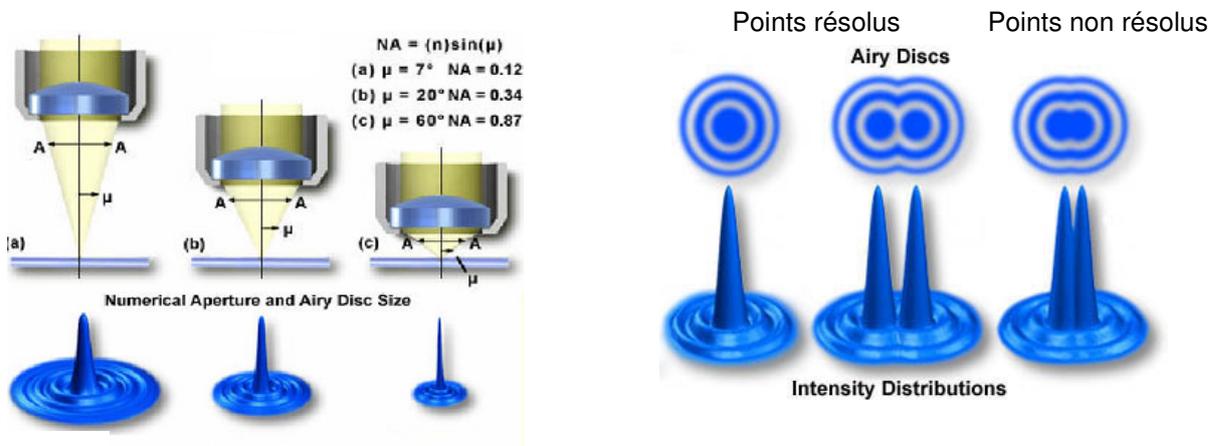
5.3 Résolution latérale et axiale

La résolution d'un microscope optique se définit comme la plus petite distance entre deux points d'un échantillon qui peuvent être distingués par l'œil ou un capteur comme deux entités différentes.

5.3.1 Résolution latérale

La résolution telle qu'elle est définie par Rayleigh correspond à la distance entre deux points objets pour laquelle le maximum d'intensité du disque d'Airy du premier point correspond au premier minimum d'intensité du disque d'Airy du second point.

Plus l'ouverture numérique de l'objectif sera grande, plus la tâche d'Airy sera étroite, et meilleure sera la résolution optique.



La résolution latérale selon le critère de Rayleigh s'exprime de la façon suivante :

$$D_{xy} = 0,61\lambda/NA$$

Où λ est la longueur d'onde d'émission et NA l'ouverture numérique de l'objectif.

5.3.2 Résolution axiale

Le même principe s'applique pour définir la résolution le long de l'axe optique Z. Selon le critère de Rayleigh, la résolution axiale est inversement proportionnelle au carré de l'ouverture numérique de l'objectif. Il est donc impératif d'utiliser des objectifs à grande ouverture numérique pour obtenir le meilleur sectionnement optique.

$$D_z = 2\lambda n/NA^2$$

Où n est l'indice de réfraction du milieu.

Il est important de noter que les résolutions latérale et axiale ne dépendent pas du grossissement de l'objectif. Un objectif 100x ne donne pas forcément une meilleure résolution qu'un objectif 63x.

6 Acquisition de l'image

6.1 Les différents types de capteurs

L'acquisition des images est réalisée par un capteur sensible à la lumière et capable de traduire ce signal lumineux en un signal électrique quantifiable.

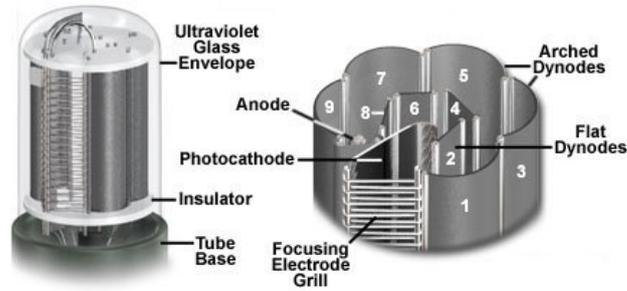
Deux types de capteurs sont utilisés en microscopie.

- Des capteurs point par point qui ne récupèrent la fluorescence émise que par un point de l'échantillon. Il est nécessaire de balayer la préparation pour former l'image d'un champ entier. C'est le cas des photomultiplicateurs utilisés en microscopie confocale à balayage monofocal (microscope confocal conventionnel).

- Des capteurs plans composés d'une matrice de capteurs élémentaires. C'est le cas des matrices CCD (Charge Coupling Device) couramment utilisés en vidéo-microscopie et composées d'une matrice de pixels.

6.1.1 Structure des détecteurs point par point

Un photomultiplicateur est composé d'une photocathode sur laquelle vient frapper le photon. L'électron créé va être amplifié en cascade grâce à plusieurs dynodes, donnant en sortie un courant électrique qui sera traduit en un signal numérique.



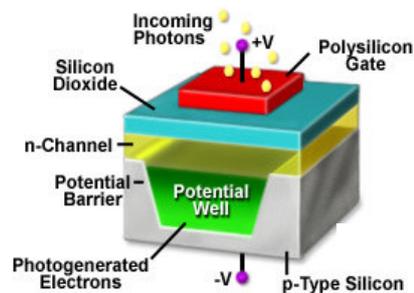
Structure d'un photomultiplicateur (PMT)

6.1.2 Structure des capteurs CCD

Chaque pixel de la matrice est composé d'une couche de dioxyde de silicium déposée sur une couche de silicium. Une couche conductrice transparente est appliquée sur cet assemblage.

Une tension positive est appliquée sur la première couche. Les électrons libres créés par les photons incidents vont être stockés dans ce qu'on appelle un puits de potentiel. Un tel puits peut contenir jusqu'à 10⁶ photons. Après accumulation des charges dans le puits de potentiel, celles-ci sont transférées vers un registre de série puis un amplificateur.

Structure d'un pixel CCD



6.2 Les différents types de capteurs CCD

Il existe 3 types de capteurs qui sont définis par leur façon de lire les charges accumulées dans les puits de potentiel : les capteurs full frame, frame transfert et interligne transfert.

6.2.1 Full frame

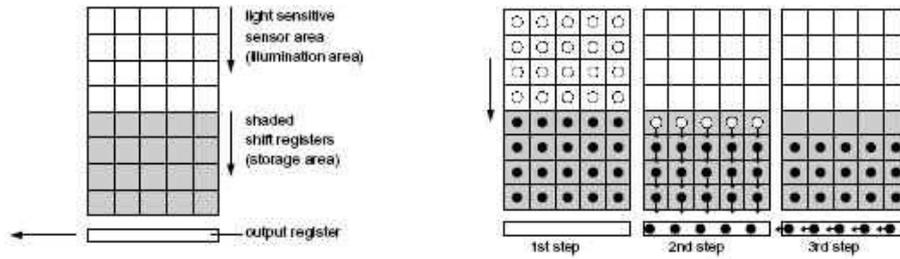
La totalité de la surface du capteur est sensible à la lumière. Après exposition, un shutter masque le photodétecteur. Les charges sont lues ligne par ligne dans le registre série.



6.2.2 Frame Transfert

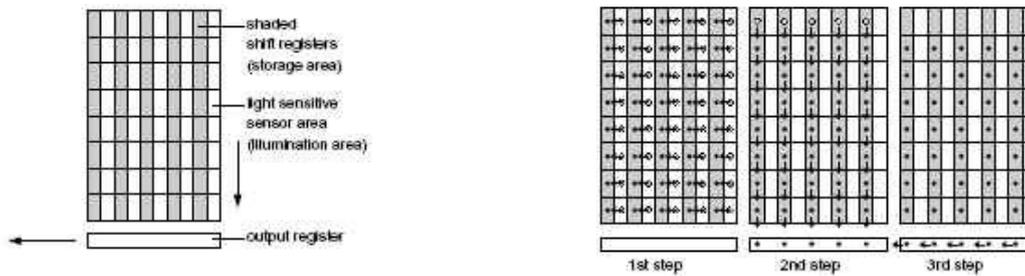
Le capteur est divisé en 2 parties : une matrice d'exposition et une matrice de stockage. Ce type de CCD n'utilise pas de shutter. Les charges accumulées dans la totalité de la surface sensible sont transférées en une seule fois vers la matrice de stockage. Le temps de transfert entre les 2 matrices est de l'ordre de quelques microsecondes.

Les charges d'une image peuvent être lues pendant que l'image suivante est acquise (mode Overlap), ce qui permet de diminuer la durée entre 2 acquisitions.



6.2.3 Interligne Transfert

Il s'agit d'une variante du frame transfert. Le capteur est subdivisé en surfaces sensibles à la lumière et en aires de stockage disposées alternativement en ligne. Les pixels accumulés dans les surfaces d'exposition sont ensuite transférés vers les aires de stockage puis lus ligne par ligne.



La surface d'exposition de ces CCD est diminuée par rapport au frame transfert. La surface sensible ne représente que 25% de la surface totale du capteur. Ces caméras sont donc moins sensibles. Les caméras les plus performantes possèdent une couche de micro-lentilles sur toute la surface du capteur permettent de concentrer la lumière sur la surface sensible (sensibilité augmentée d'un facteur 2).

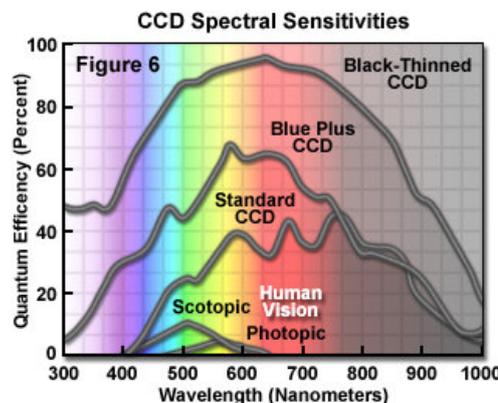
6.3 Caractéristiques des capteurs CCD

6.3.1 Sensibilité spectrale

La sensibilité d'un capteur CCD représente la plus petite intensité lumineuse pour laquelle il pourra fournir un signal supérieur au bruit de mesure. On distingue la sensibilité absolue (nombre d'électrons détectables) de la sensibilité relative (nombre d'électrons permettant de passer à un niveau d'intensité supérieur).

La sensibilité du capteur dépend de la probabilité qu'un photon incident produise un électron. On parle d'efficacité quantique. Les caméras utilisées en imagerie scientifique possèdent une efficacité quantique de l'ordre de 60%.

Cette efficacité quantique dépend de l'énergie du photon. Elle est donc directement liée à la longueur d'onde des photons détectés. La sensibilité spectrale d'un capteur dépend du matériau utilisé pour piéger les électrons. La plupart des capteurs sont en silicium, qui permet d'obtenir une bonne sensibilité dans le visible et l'infrarouge.



Sensibilité spectrale de différents types de CCD

6.3.2 Rapport signal/bruit

Lorsqu'une mesure de signal est effectuée plusieurs fois à intensité lumineuse reçue constante, la valeur du signal oscille en fonction du temps avec une amplitude qui correspond au bruit global de la mesure.

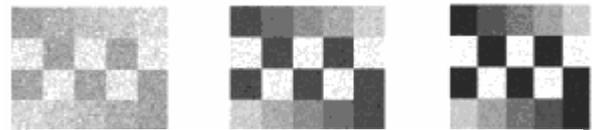
On distingue trois sources de bruit :

- le bruit au noir, qui correspond à la production de charges par le pixel même lorsque celui-ci ne reçoit aucune illumination. Pour réduire ce bruit, les caméras sont refroidies à -20°C ou -30°C . Le bruit au noir devient alors négligeable.
- le bruit de lecture ou bruit électronique, qui est indépendant du temps d'exposition mais augmente fortement si la cadence de lecture augmente. Il est dû à des incertitudes de mesure réalisées lors de la quantification du signal par la CCD. Il suit une distribution Gaussienne.
- le bruit photonique dû à la variation naturelle du flux de photons. La distribution des photons collectés par une CCD suit une loi de Poisson. Ce bruit est estimé comme la racine carrée du signal.

Le rapport signal/bruit (SNR : Signal to Noise Ratio) peut être mesuré de la façon suivante :

$$\frac{IQEt}{\sqrt{(IQEt + Ndt + Nr^2)}}$$

- I = Flux de photons (photons/pixel/second)
- QE = Efficacité quantique
- t = temps d'intégration (secondes)
- Nd = bruit au noir (électrons/pixel/sec)
- Nr = bruit de lecture (électrons)



Variation du rapport signal sur bruit

6.3.3 Résolution spatiale

La résolution spatiale de la caméra CCD est directement liée à la taille des pixels. La taille du pixel varie de $6.45\mu\text{m}$ pour les caméras les plus performantes, à une dizaine de micromètres.

Si l'intensité du signal récolté est trop faible, il y a possibilité de regrouper les pixels 2 par 2, 4 par 4, ... C'est le binning. En plus d'augmenter l'intensité du signal récolté et donc le rapport signal/bruit, cette méthode permet également d'augmenter la vitesse de lecture et d'obtenir des images moins lourdes en terme de stockage de fichier. Cependant, tout ceci se fait au détriment de la résolution : le binning réduit l'intervalle d'échantillonnage en XY.

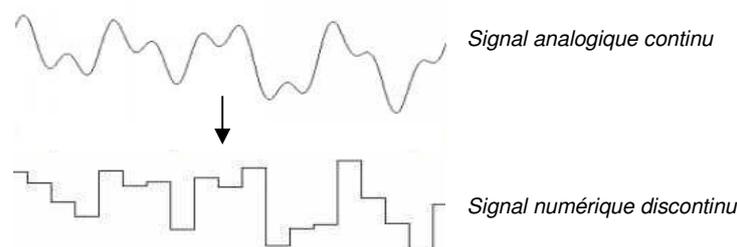
L'intervalle d'échantillonnage (taille du pixel caméra/grossissement objectif) est défini par le critère de Nyquist (résolution optique divisée par 2).

6.3.4 Dynamique de mesure et de numérisation

La dynamique de mesure correspond au nombre de niveaux d'intensité lumineuse différents que le capteur peut distinguer, entre un signal nul et un signal d'intensité maximum. La plupart des caméras codent sur 212 niveaux de gris (4096 niveaux).

Le signal formé par le capteur est analogique, c'est à dire que sa variation est continue entre les valeurs minimales et maximales. Après avoir été lues, les charges sont transférées vers un convertisseur analogique-numérique qui convertit les charges (en volt) en valeurs binaires afin que le signal puisse être interprété par l'ordinateur. A chaque pixel correspond donc une valeur numérique qui représente l'intensité du signal.

La numérisation doit être effectuée à une dynamique au moins équivalente à celle de la mesure.



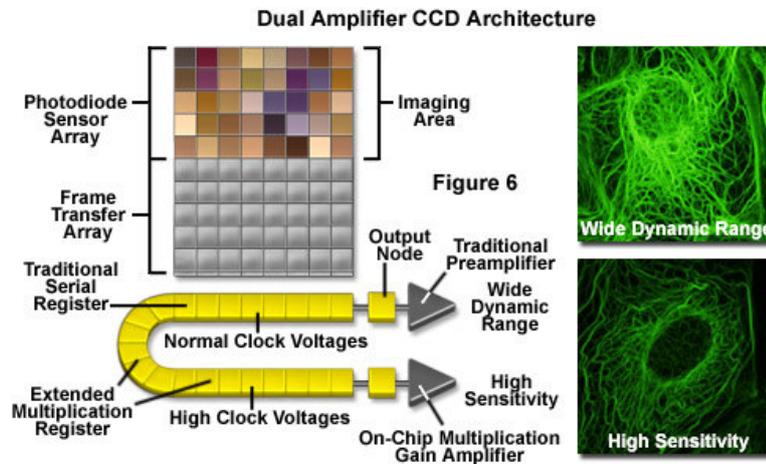
6.4 Les caméras à amplification de signal

6.4.1 La technologie EMCCD

Les EMCCD (CCD à multiplication d'électrons) utilisent le concept classique de fonctionnement des CCD. Une structure à multiplication d'électrons (registre de gain) est insérée entre le registre de série et l'amplificateur. Le registre de gain est constitué de plusieurs paires d'électrodes. Une forte tension est appliquée entre ces 2 électrodes, créant au moment du transfert des électrons une accélération et un impact d'ionisation. Cet impact génère alors de nouveaux électrons, proportionnellement au potentiel

appliqué sur les électrodes. La probabilité au niveau d'un couple d'électrodes est relativement faible (de 1 à 1.6%) mais en répétant cette opération un grand nombre de fois, un gain important peut être obtenu (x1000).

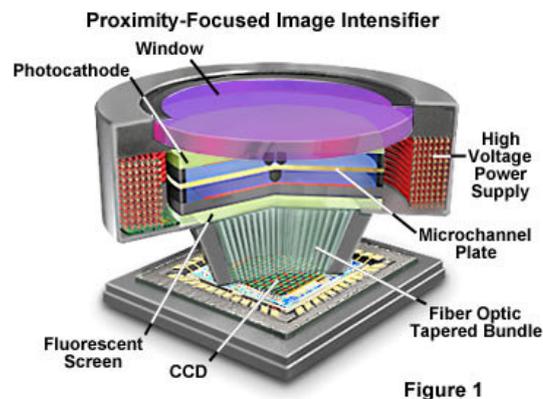
En insérant ce registre de gain avant l'amplificateur de sortie, le signal est amplifié par rapport au bruit de lecture, ce qui tend à augmenter le rapport signal/bruit.



6.4.2 Les CCD intensifiées

Ces caméras sont utilisées pour des applications en très basse lumière nécessitant une bonne résolution temporelle.

Un intensificateur d'image est placé devant le capteur. Il s'agit d'un tube sous vide comprenant une photocathode et des galettes de microcanaux (MCP : Micro Channel Plate) permettant d'amplifier les électrons qui iront frapper un écran de phosphore, provoquant alors la génération de lumière. Les photons générés sont focalisés sur le capteur CCD par une lentille ou par fibre optique. Le couplage par fibre est bien meilleur, permettant de récolter 60% de lumière en plus.



6.5 Les caméras CCD couleurs

Ces capteurs ne sont sensibles qu'à l'intensité lumineuse. Pour intégrer la couleur ; il faut placer devant le capteur des filtres qui feront que les pixels de la matrice ne seront sensibles qu'à une gamme de longueurs d'onde donnée.

Il existe quatre types de caméras CCD couleur :

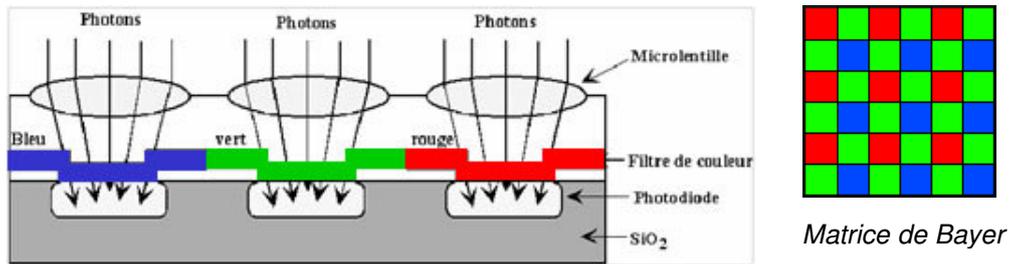
- les caméras à matrice de Bayer
- les CCD monochromes avec roue de filtres ou filtre de cristaux liquides
- les tri-CCD
- les CMOS trois couches

6.5.1 Les CCD à matrice de Bayer

Chaque pixel du capteur est recouvert par un filtre miniature rouge, vert ou bleu. Le système de matrice de Bayer utilise une cellule de 4 pixels comprenant 2 pixels verts disposés en diagonale, l'autre diagonale comprend un pixel rouge et un pixel bleu. Cette répartition correspond à la sensibilité de notre vision.

L'image est acquise en une seule fois. La reconstitution des couleurs se fait par interpolation des couleurs manquantes à partir des pixels voisins.

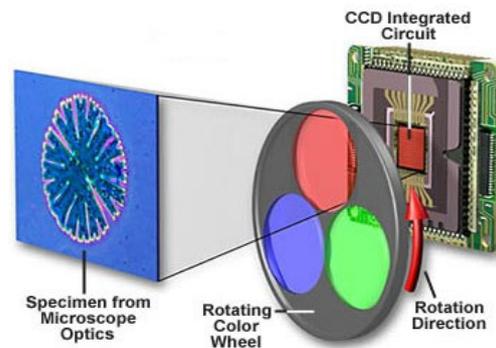
Le canal vert est appelé canal de luminance parce qu'il contient les informations de luminosité de chaque partie de l'image. Les canaux rouge et bleu sont les canaux de chrominance, ce sont eux qui sont responsables de la couleur sur l'image.



6.5.2 Les CCD monochromes avec roue de filtres ou filtre de cristaux liquides

Une roue de filtres (vert, rouge, bleu et neutre) pilotée par ordinateur est située devant le capteur CCD monochrome. L'image couleur est acquise séquentiellement. C'est la solution la plus couramment utilisée en microscopie de fluorescence. C'est un bon compromis entre résolution, sensibilité et rapidité.

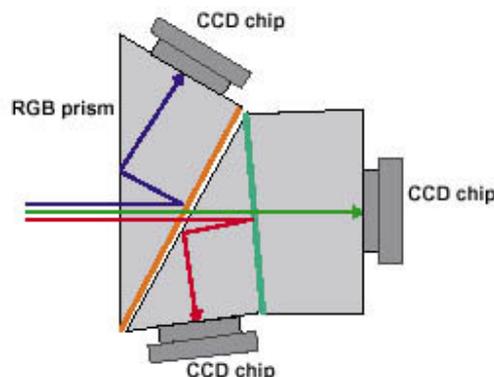
Les CCD monochromes avec filtre de cristaux liquides fonctionnent sur le même principe. L'image couleur est acquise séquentiellement, afin d'ajuster la bande spectrale pour chaque canal.



6.5.3 Les tri-CCD

Ce type de caméra utilise un diviseur de faisceau composé de trois miroirs dichroïques. La lumière est donc divisée en trois composantes spectrales qui sont dirigées sur trois capteurs CCD indépendants.

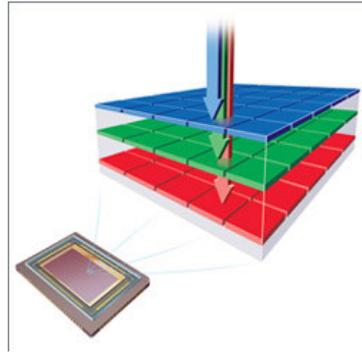
Ce type de caméra est moins sensible que les CCD monochromes avec roue de filtres. Elle peut être utilisée en microscopie de fluorescence avec des signaux forts.



6.5.4 Les CMOS 3 couches

Ce type de capteur est basé sur une technologie récente qui consiste à utiliser les propriétés spectrales d'absorption de la lumière à différentes profondeurs de silicium. Le capteur est composé de 3 couches de

silicium. Chacune de ces couches absorbe une bande spectrale différente. L'image couleur est donc acquise en une seule fois.



Structure d'une CMOS 3 couches (Foveon)

7 Dégradation de l'image

7.1 Sources de dégradation

Les sources de dégradation d'une image sont multiples. Elles peuvent provenir de l'objet étudié lui-même, du microscope ou bien du capteur utilisé pour faire l'image.

7.1.1 Au niveau de l'objet

Les objets diffusent et absorbent à la fois la lumière d'illumination et la lumière émise, ce qui induit une perte de lumière.

Pour les objets fluorescents, une intensité d'illumination trop intense induit le phénomène de bleaching. Lors d'une acquisition intégrée sur un temps long, on observera une diminution du rapport signal/bruit.

7.1.2 Au niveau du microscope

La dégradation principale provient de la diffraction des rayons lumineux par les lentilles du système optique.

Des défauts dans les lentilles peuvent également induire des distorsions géométriques ou de sphéricité.

La qualité de l'illumination est très importante. Une illumination non homogène induira des différences d'intensité sur l'image.

7.1.3 Au niveau du capteur

Le signal produit par le capteur est altéré par du bruit (au noir, de mesure ou de lecture).

7.2 Correction des dégradations

Parmi toutes ces dégradations, certaines peuvent être modélisées : bruit photonique (gaussien), bruit électronique (loi de poisson), diffraction (fonction de transfert de contraste ou PSF), bleaching (diminution exponentielle de l'intensité lumineuse en fonction du temps).

Un traitement mathématique inverse peut en partie les corriger.

Les autres types de dégradations doivent être évitées au mieux : optimisation de l'alignement des lampes afin d'obtenir une illumination du champ homogène, bon échantillonnage spatial, bonne dynamique de mesure et de numérisation, choix du milieu d'immersion, objectifs corrigés ...

Ressources utiles :

<http://www.microscopyu.com/>

Cours « Introduction aux caméras CCD », Philippe Rochon.

Les formats des fichiers images

Table des matières

1. Définition d'un format de fichier	2
2. Images Bitmap, vectorielles et méta-formats	3
2.1 Vectoriel	3
2.2 Bitmap	3
2.3 Méta-formats	4
3 Les couleurs	4
3.1 Dynamique ou profondeur de couleur	4
3.2 Remarque sur la taille d'une image	5
3.3 Le codage de la couleur	5
3.4 La transparence simple et la couche ALPHA	5
4 Extensions MS-DOS, Types McIntosh, En-têtes de fichiers.	6
5 Format compressés ou non compressés	7
6 Les principaux formats de fichiers utilisés actuellement pour les images numériques	8
6.1 Formats les plus courants.....	8
6.2 Formats compressés avec perte d'information	11
6.3 La vidéo numérique	12
6.3.1 Les codecs	12
6.3.2 Les formats	13

1. Définition d'un format de fichier

Une image numérique se mesure en pixels ; le pixel n'est pas une unité de mesure arbitraire, il signifie (un peu d'étymologie ...) PICture ELement, donc le plus petit élément d'une image numérique non-vectorielle (ou raster ou bitmap), c'est véritablement l'atome de l'image au sens de DÉMOCRITE. On ne peut pas le détailler, il n'a qu'une seule et unique valeur (nous verrons plus loin ce que cela signifie).

Le format de fichier correspond à la manière dont les valeurs des pixels d'une image sont ordonnés dans un fichier informatique (généralement un fichier binaire mais ça peut être dans certains cas un fichier ASCII¹).

Dans un fichier graphique, on trouve le plus souvent à la fois des données graphiques (couleur des pixels par exemple) et des données non graphiques (informations complémentaires nécessaires à la restitution de l'image telles que, par exemple, ses dimensions, le nom du programme créateur, des annotations de l'utilisateur, etc ...). Il existe plusieurs dizaines de formats s'appliquant aux images numériques : beaucoup de ceux qui ont été développés au fil des années ont l'inconvénient majeur d'être des formats propriétaires qui sont (volontairement ou non) pas ou mal documentés; ces formats sont aujourd'hui sinon obsolètes du moins très peu utilisés.

Le choix d'un format de fichier est important si on veut s'assurer de pouvoir relire les images sauvegardées plusieurs années après leur enregistrement et cela sur différents systèmes informatiques et avec divers logiciels : la logique pousse alors l'utilisateur à utiliser un format standard, c'est-à-dire un format de fichier qui a fait l'objet d'une norme et dont les spécifications sont précises et accessibles à tous. Les formats standards s'appliquant aux images sont alors beaucoup moins nombreux.

Les différentes catégories d'images et types de formats associés : on peut sommairement classer les fichiers d'images (et donc les formats de fichiers associés) en plusieurs catégories : les images bitmap et les vectorielles. Les formats de fichiers bitmap existent sous forme compressée ou non. Les images bitmap peuvent être soit fixes soit des séquences vidéo : les fichiers vidéo étant très importants en taille, des formats particuliers, toujours basés sur une compression des données avec perte d'information, sont utilisés. D'autre part, on peut distinguer les formats de fichiers non compressés et les formats faisant intervenir une compression des données qui peut induire ou non une perte irréversible d'information; on a donc trois possibilités de codage des images :

- Pas de compression
- Compression sans perte d'information
- Compression avec perte d'information

¹ASCII = American Standard Code for Information Interchange

Ces trois possibilités sont applicables aux différents formats de fichiers images quel que soit leur type.

Un fichier image contient, outre les valeurs des pixels, un en-tête qui décrit son type, un en-tête d'information qui décrit comment les valeurs sont organisées dans le fichier (et donc comment elles doivent être interprétées lors de la lecture). Dans certains cas, on peut trouver une palette des couleurs utilisées dans l'image, et également un « footer », c'est-à-dire une zone, située à la fin du fichier, qui peut contenir diverses informations (annotations de l'utilisateur par exemple), cas des fichiers des anciens microscopes confocaux Biorad©.

Sur les systèmes DOS, Windows et OS/2, les formats de fichiers sont désignés par l'extension du nom de fichier (ou suffixe) (cette extension n'est pas affichée par défaut dans Windows : il faut modifier les options d'affichage des noms de fichiers pour la faire apparaître). Sur Mac OS, les fichiers sont caractérisés d'une autre manière (ressource) mais rien n'empêche de nommer les fichiers en rajoutant une extension : cela facilitera les échanges entre les environnements MAC et PC (sous Unix et Linux on peut respecter également, sans y être forcé, la tradition de l'extension pour les format de fichiers et cela pour la même raison que précédemment).

2. Images Bitmap, vectorielles et méta-formats

2.1 Vectoriel

Les images vectorielles sont des représentations conceptuelles des formes, issues notamment des logiciels de dessin industriel (et plus généralement de tous les logiciels de dessin vectoriel); le format vectoriel est aussi utilisé pour définir les polices de caractères (PostScript, TrueType). Dans ce mode, une ligne, un cercle ou un rectangle sont définis par 2 points, un triangle par 3 : les formes sont des fonctions mathématiques. La résolution est donc variable en mode vectoriel puisque quel que soit le zoom que vous faites dans votre image, tous les composants sont recalculés.

Ces formats sont très souples pour les déformations géométriques et les changements d'attributs (couleur, épaisseur, alignement, ...), les objets sont modifiables indépendamment les uns des autres.

Malheureusement, la nature des capteurs utilisés en imagerie ne nous permet jamais d'utiliser de telles images (quel dommage ! Des images qu'on pourrait zoomer à l'infini, le rêve du microscopiste !!!).

2.2 Bitmap

Les images bitmap (ou raster) sont pixellisées (photos numériques, images issues de scanner, caméras CCD, microscope confocal et la plupart des systèmes d'acquisition

d'images). L' image est un ensemble de points juxtaposés, ordonnés en lignes et en colonnes; chaque point peut être localisé en abscisse et en ordonnée (le point d'origine se trouvant généralement en haut à gauche). Les formats bitmap rendent compte des valeurs des pixels dans l'image : dans le fichier, le codage des pixels se fait de manière mono dimensionnelle (pour une image de 640 x 480 pixels, le pixel n° 650, par exemple, occupe dans l'image, la dixième position de la deuxième ligne).

2.3 Méta-formats

Il existe enfin de (trop) nombreux formats intelligents qui contiennent d'autres informations comme le format *stk* de Metamorph© qui contient les paramètres de l'acquisition et TOUTES les images d'une pile d'image. On parle alors de méta-fichiers (metafiles) pour les décrire.

3 Les couleurs

3.1 Dynamique ou profondeur de couleur

La profondeur de couleurs est le nombre maximal de couleur que peut avoir une image. On parle de 1, 2, 8, 16, 24 bits ou plus.

Une image 1 bit (2^1) est une image dont chaque pixel ne peut avoir que 2 valeurs: 0 ou 1. On ne s'en sert que pour les fax ou le scan de texte. Le 2 bits (2^2) n'est pas utilisé. Le 4 bits (2^4) permet 16 combinaisons : 0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111, 1000, ..., 1111.

Pour des raisons pratiques, on peut les coder en hexadécimal (base 16) : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Le 8 bits (2^8) permet 256 niveaux de couleur (ou de gris) de 00000000 à 11111111 ce qui donne 00 à FF en hexadécimal. Le format GIF est codé sur 8 bits. Le 10 bits n'est plus utilisé que par de vieux scanners, le 12 bits (4096 niveaux) est une profondeur classique des caméras CCD utilisées en microscopie. Le 16 bits permet 65536 niveaux (2^{16}) est le format d'enregistrement de bien des logiciels de microscopie (Metamorph© , par ex.). Le 24 bits permet 16 777 216 niveaux. Il s'agit souvent en fait d'une image couleur dont les 3 composantes R(ouge), V(ert) et B(leu) sont codées sur 8 bits. Les images JPEG sont codées en 24 bits. Le 30 bits (chaque couleur sur 10 bits) est courant pour les scanners de bureau. Le 32 bits désigne souvent une image destinée à l'impression C(yan), M(agenta), Y(ellow), K (:noir). Seulement simulé par l'ordinateur qui utilise des tables de correspondance entre RVB (son mode natif) et CMYK, cet espace colorimétrique est réservé aux imprimeurs qui procèdent par recouvrement de pigments. Le 32 bits désignent de plus en plus des images 24 bits auxquelles est ajoutée une couche dite « alpha » qui contient des informations de transparence de l'image.

3.2 Remarque sur la taille d'une image

Pour calculer la taille d'une image en kilo-octets (Ko), on utilise la relation :

- Poids en bits = nombre de pixels (hauteur x largeur) x profondeur de couleur
- Poids en octets = Poids en bits / 8. => poids en Ko = Poids en octets / 1024

Ex. : $512 \text{ pixels} * 512 \text{ pixels} * 16 \text{ bits} / 8 = 524288 \text{ octets}$ soit 512 Ko. Ce chiffre ne tient pas compte de l'en-tête de fichier ou des spécificités du format (compression, ...)

3.3 Le codage de la couleur

Il existe deux manières très différentes de traiter la couleur numérique : le mode True Color et les Couleurs Indexées. En true color (couleurs vraies), chaque pixel peut prendre une valeur parmi 16 millions (16777216); cette valeur est alors codée sur 3 fois 8 bits dans le système RVB, par exemple R=255,V=0,B=0 représentera le rouge le plus absolu. Bien sûr, on peut aussi trouver des images avec des composantes colorées codées sur plus de 8 bits ou dans un autre espace colorimétrique (systèmes HSV, CMYK, ...). Le système RGB (ou RVB en français) est le plus utilisé en imagerie numérique. Les imprimeurs préfèrent le CMYK (CMJN en français pour Cyan, Magenta, Jaune, Noir).

En couleurs indexées, le fichier est séparé en deux parties distinctes; d'une part la table des couleurs ou palette (en anglais LUT pour LookUp Table = table de correspondance des couleurs) qui sert d'index, et de l'autre, plutôt que d'être spécifiée directement, la valeur d'un pixel va désigner une entrée dans la palette laquelle identifie une couleur (ou un niveau). Ceci permet de diminuer la taille du fichier.

3.4 La transparence simple et la couche ALPHA

Il existe deux modes de transparence d'images. La plus basique des deux, qu'on appellera transparence simple est utilisé dans les formats d'images en couleurs indexées, comme le GIF : on décrète qu'un des éléments de la table de couleurs n'est pas une valeur colorée mais au contraire une absence de couleurs, qu'on pourrait comparer au trou d'un pochoir. La couche alpha est un peu plus intéressante car elle permet une approche graduée (progressive) de la transparence. En plus des 8 bits de Rouge, 8 de Bleu et 8 de Vert, l'image contient une couche en niveau de gris, donc 8 bits pour coder la transparence.

4 Extensions MS-DOS, Types McIntosh, En-têtes de fichiers.

Pour connaître la nature d'une image, les logiciels et les systèmes d'exploitation ont plusieurs méthodes. La moins évidente et la plus complète est celle de Mac OS jusqu'à la version 9, qui sépare chaque fichier (image ou autre d'ailleurs) en deux parties distinctes, le *data fork* et le *resource fork*. Le *data fork* contient les données proprement dites et le *resource fork* divers attributs (type, créateur, date, ...) et objets (icônes, sons, etc ...). Tout fichier Mac est donc un meta-fichier en puissance, l'inconvénient est la place sur le disque : chaque fichier occupera au minimum 2 clusters (plus petit élément de stockage). Un fichier texte de 1Ko occupera 64 Ko (les clusters font en moyenne 32 Ko sur les gros disques). Tout fichier mac a un créateur et un type. Le type est par exemple TEXT pour du texte ou du HTML; le créateur est le logiciel qui a créé ce fichier qui sera automatiquement ouvert lorsqu'on ouvre un fichier en cliquant sur son icône. Ces ressources peuvent être éditées à l'aide du logiciel *resedit* (à manipuler avec extrême précaution). Il est à noter que la famille Mac OS X (dix mais X pour Unix©) utilise les attributs étendus de fichiers (gestion des droits, lecture des *magic numbers*, ...) comme tout bon système d'exploitation qui se respecte.

Sous Windows, c'est le nom du fichier et plus précisément son suffixe qui permet de savoir de quel type est le fichier (spéciale dédicace aux utilisateurs Mac qui envoient des pièces jointes sans extension dans leurs mails à des gens sous PC); ainsi le fichier image.tif est un fichier TIFF. La correspondance entre l'extension et le fichier créateur est stockée dans des fichiers systèmes comme *win.ini* ou dans la base de registres. Pour modifier ces relations, on peut aller dans n'importe quelle fenêtre dans Affichage / options des dossiers / Types de fichiers. Vous pouvez aussi passer par la base de registre en tapant "*regedit*" dans démarrer / exécuter mais je ne cautionne pas cette manipulation.

Enfin, il existe les types MIME (Multipurpose Internet Mail Extensions), indépendants de la plate-forme, créés pour l'échange de fichiers de fichiers multimédias et l'usage de jeux de caractères étendus via e-mail. Chaque type mime appartient à un groupe : image, application, audio, vidéo, etc ... Les types en "x" (image/x-pict par exemple) ne sont pas universels et leur portabilité implique une entente préalable entre les logiciels qui échangent ces données. La nomenclature mime rencontre un grand succès du fait de sa simplicité. L'essor du Web laisse penser que ce sera bientôt le standard du typage de fichier.

Comment déterminer le format qui a été utilisé pour enregistrer un fichier ?

- Se renseigner auprès de celui qui a fourni le fichier !
- Lire l'en-tête de ce fichier : La lecture de l'en-tête peut se faire à l'aide d'un éditeur de texte (notepad, simple text, bbedit, emacs, vi) ou d'un éditeur hexadécimal. Parfois, l'en-tête est lisible en clair : GIFS7a (en hexadécimal : 474946383761h) et Gif89a (474946383961h) sont par exemple les en-têtes des fichiers GIF. Pour les JPEG et la majorité de ses variantes et des formats qui ont recours à son mode de compression,

JFIF, PICT et parfois PSD, les trois premiers octets sont FFD8FFh (le quatrième varie d'un logiciel à l'autre). Les huit premiers octets du format PNG sont 89504E470D0A1AOAh. Les quatre premiers octets d'un fichier TIFF sont 4D4D002Ah ou 49492A00h. Les fichiers BMP contiennent le mot BM ou le mot BM6. Les 18 derniers octets d'un fichier TGA sont : TRUEVISION-XFILE.\0 (sauf dans le cas des TGA version 1). Les fichiers EPS ont un en-tête très complet et écrit en clair qui comprend le nom du fichier, son type et le logiciel qui l'a créé.

5 Format compressés ou non compressés

On a souvent intérêt à enregistrer dans un format compressé. Il existe deux formes radicalement opposées de compression : les compressions sans perte de qualité (Huffman; Lempel, Ziv, Welch -LZW-; RLE -Run Length Encoding-) et les compressions avec pertes (JPEG; MPEG 1 et 2).

Une des méthodes de compression sans perte de qualité est de supprimer les informations redondantes. C'est ainsi que fonctionne la compression RLE (Run Length Encoding) utilisé par TGA ou PSD : une ligne noire de 10 pixels N N N N N N N N N N se codera 10 x N , rentable ! Par contre, s'il y a un grand nombre de valeurs peu différentes (un dégradé par exemple) : A B C D E F G H I J sera compressé en 1A 1B 1C 1D 1E 1F 1G 1H 1I 1J ce qui nous fait plutôt perdre de la place. Ce type de compression est donc surtout réservé aux grands aplats de couleurs comme dans les logos, les titres, ...

Pour le même type d'usage, la compression LZW fonctionne un peu différemment. Elle code l'image sous forme de motif, associés à un dictionnaire. Par exemple, le motif qQZERGergqRGqerGQer dans l'image revient assez souvent (je sais, ça m'étonne aussi) et sera associé à 1 dans le dictionnaire, c'est-à-dire qu'à la décompression, chaque fois que 1 sera rencontré dans le fichier on remplacera par qQZERGergqRGqerGQer mais me direz-vous , comment savoir si le 1 rencontré doit être remplacé ou fait partie de l'image ?

Plusieurs méthodes sont possibles, soit on utilise des caractères qui ne seront pas dans l'image pour construire le dictionnaire soit on n'en a qu'un par exemple c ; ainsi c1 sera remplacé par qQZERGergqRGqerGQer c2 par un autre motif, etc ... Ce mode de compression est utilisé dans les formats TIFF v.5, GIF, PDF et PostScript 2.

Enfin, la compression JPEG est destructive. On la retrouve dans JPEG, PICT (et oui ...) certains TIFF v.6 (car on peut choisir la compression). Le jpeg profite du peu de sensibilité de notre œil aux variations de tons (il est surtout sensible aux différences de luminosité) pour appauvrir les images sans que nous puissions le voir. L'algorithme utilisé décompose l'image en blocs et code les variations de tons, vues comme oscillations périodiques, plutôt que les tons eux-mêmes. Les variations brusques, correspondant à de grandes fréquences seront peu voire pas codées, qu'importe puisqu'elles décrivent le plus souvent du bruit ! Ce type de compression est donc adaptée aux images constituées de tons proches et progressifs comme vos photos de vacances de cet été, fond gris uniforme ;-)

Bien d'autres formes de compression avec perte d'information existent et sont souvent

liées aux types de données. Pour obtenir, si nécessaire, des taux de compression très importants, ces méthodes essaient de minimiser les effets des pertes d'information par différentes stratégies qui vont s'appliquer à un type de fichier particulier (MP3 pour le son, JPEG pour les images fixes, MPEG pour les films).

6 Les principaux formats de fichiers utilisés actuellement pour les images numériques

6.1 Formats les plus courants

On s'attachera à décrire *rapidement* les formats non compressés, compressés sans perte d'information et multi-résolution

TIFF (Tag Image File Format), (extension .TIF), images Bitmap très populaire dans l'édition, il fut créé par Aldus (qui appartient maintenant à Adobe) et Microsoft (encore lui), le TIFF a été conçu du départ pour l'acquisition et la création d'images en vue de l'impression et est donc très souvent proposé comme format par défaut des logiciels de numérisation d'images (les fichiers stk de Metamorph© ne sont que des piles de TIFF encapsulés). Il est indépendant de la plate-forme sur laquelle il a été créé et est compatible avec lui-même : une image TIFF peut être lue par n'importe quel logiciel qui supporte ce format (modulo la compression) et les différentes versions et révisions (nous en sommes à la 6ème) sont compatibles entre elles, un vieux logiciel est presque toujours capable de lire un TIFF récent.

On s'étonnera à ce sujet de l'opportunité du choix qui est donné lorsqu'on enregistre une image en TIFF dans photoshop de le faire au format Mac ou PC, cela ne change que l'en-tête et donc l'ordre des octets 4D4D pour un en-tête Mac (les gros indiens, octets de poids fort d'abord) et 4949 pour les PC (petits indiens, octet de poids faible d'abord). Ce format peut gérer quasi toutes les profondeurs de couleurs : niveaux de gris, 16, 24 ou 32 bits. Pour être visionné de la même façon sur des systèmes différents, il contient des informations sur le gamma.

En fait ce format supporte ou non une compression sans perte de données du type LZW (LEMPERL-ZIV-WELCH), Packbits, HUFFMAN, CCITT 3 et JPEG (la compression la plus utilisée est la compression LZW pour les images en couleur; les formats CCITT 3 et HUFFMAN ne traitent que des images en noir et blanc). Le format TIFF est préférentiellement utilisé en mode non compressé car on s'assure de cette manière une compatibilité quasi universelle avec tous les logiciels d'imagerie et cela quelle que soit la plate-forme matérielle utilisée. Le format TIFF s'applique à tous les types d'images 2D (noir et blanc, niveaux de gris, palette de couleurs, couleur 24 bits) et supporte les images multiples comme les stacks Metamorph dans la limite de 4Go et est à l'heure actuelle le format le plus répandu et utilisé.

Type Mac : *TIFF* Extension Windows : *.tif* MIME : *image/tiff*

BRUT (RAW DATA) (extension .RAW), images Bitmap. Ce format (qui en fait n'en est pas réellement un) consiste à ne coder que les valeurs de pixels dans un fichier binaire; il faut que l'utilisateur connaisse la taille de l'image (en X et en Y) et le nombre de bits utilisé pour le codage pour pouvoir relire une image enregistrée de cette manière. Ce format s'applique principalement aux images en niveaux de gris bien qu'il soit possible de l'utiliser pour des images en mode RVB. L'emploi de ce format n'est pas très pratique mais il peut s'avérer utile pour des échanges inter plate-formes quand aucun format commun n'est disponible. Seuls certains logiciels comme, par exemple, NIH-Image ou Scion-Image et Adobe Photoshop proposent une lecture des fichiers au format brut. L'écriture au format brut est proposée par un nombre encore plus restreint de logiciels: Photoshop par exemple.

EPS (Encapsulated PostScript), (extension .EPS), tous les types d'images

Ce format s'applique bien aux images destinées à l'impression sur des imprimantes laser; il est basé sur le langage de description de page Adobe PostScript et a été popularisé par les imprimantes laser Apple LaserWriter. Un fichier EPS est généralement constitué de deux parties : une partie texte (langage PostScript) servant à l'impression, et une partie (optionnelle) contenant une image bitmap au format PICT pour l'affichage sur écran. Ce format ne se transfère pas très bien entre les différentes plate-formes : souvent le fichier transféré n'est plus éditable (une exception : les fichiers EPS d'Adobe Illustrator). Idée intéressante au départ, mélangeant objets vectoriels et bitmap, codé en binaire ou ASCII (donc visible avec un traitement de texte), sa partie bitmap peut être compressée en JPEG, il peut contenir une pré-visualisation TIFF; ce format a tellement dérivé en versions toutes différentes et incompatibles entre elles qu'il n'est pratiquement plus utilisé.

PICT (MacIntoshMcIntosh Picture ou QuickDraw Picture) C'est le format natif du Mac et est supporté par tous les logiciels de cette plate-forme (y compris les traitements de textes, les bases de données, etc...) . Couvre toutes les profondeurs de couleur jusqu'à 24 bits et peut utiliser la compression JPEG ou Packbits (à choisir car non-destructive). Sa complexité (gestion des appels quickdraw) n'en fait pas un bon format multi-plate-forme, même si beaucoup de logiciels Windows le gèrent efficacement. Type Mac : PICT Extension Windows : .pic .pct Type MIME : image/pict

DXF (extension .DXF), images vectorielles Format standard des logiciels de dessin vectoriel (AutoCAD, Autodesk); s'applique aux images 2D et 3D. Sa particularité est d'être codé en texte (ASCII), au lieu du codage binaire et donc d'être lisible avec un simple traitement de texte : ceci le rend quasiment universel.

BMP (Bitmap) avec compression RLE, (extension .BMP), images Bitmap De part son ancienneté, ce format est reconnu par tous les logiciels de retouche d'images sur PC; il est par contre assez mal géré sur Mac.

TIFF-LZW (extension .TIF), images Bitmap Description : voir ci-dessus. On remarquera que l'extension ne permet pas de savoir si le fichier est ou non compressé. Cette information est inscrite dans l'en-tête du fichier (Header).

GIF (Graphics Interchange Format), (extension .GIF), images Bitmap, animations Le format

GIF a subi plusieurs évolutions qui l'a conduit à devenir un des deux principaux standards du Web. Depuis longtemps orienté petite taille, en kilo-octets (couleurs indexées et compression LZW), le format GIF intègre la possibilité d'une transparence simple, un affichage progressif (entrelacement) et des possibilités d'animation (GIF89a). Il est parfait pour les aplats et déconseillé pour l'impression, de la même façon, étant en couleurs indexées, seuls 256 niveaux de gris sont disponibles ce qui est insuffisant pour quasi tous les équipements scientifiques. Type Mac : GIF Extension Windows : .GIF Type MIME : image/gif

PNG (Portable Network Graphic - ou png's not gif). Ce format PNG (prononcez ping pour faire informaticien ;-)) a été créé pour remplacer le format GIF du fait des limites de ce dernier, mais surtout pour des questions de droits : tout éditeur de logiciels qui souhaite manipuler des GIF est désormais astreint à verser des royalties à UNISYS, propriétaire de l'algorithme LZW (et oui !). Il n'est pas supporté par les vieux navigateurs ou partiellement. Le PNG est un format compressé sans pertes mais avec un excellent rendement : les images sont 10 à 30% plus légères que les GIF ! En couleurs indexées, le PNG peut faire 1,2,4 et 8 bits comme le GIF. En niveaux de gris, il peut monter en 16 bits et en True Color, avec 8 ou 16 bit/couche, on a accès à du 24 ou 48 bits. Sa couche Alpha se code sur 8 ou 16 bits ce qui est plus intéressant que le GIF (1 seul niveau d'absence). Il contient comme le TIFF des informations sur la correction gamma et cerise sur le gâteau permet de détecter si un fichier est endommagé avant de l'ouvrir. Type Mac : PNGf Extension Windows : .png Type MIME : image/png

TGA (Targa : Truevision Advanced Raster Graphic Adapter) Un des plus grand classiques de l'image RVB (destiné à l'écran donc ...). Utilisé depuis longtemps en vidéo ou sur des consoles graphiques (nostalgique de l'*Amiga*, je t'ai reconnu), il supporte toutes les profondeurs du 1 au 32 bits, peut avoir une couche Alpha et foules d'autres informations, pour réduire sa taille, supporte la compression RLE et les couleurs indexées.

WMF (Windows metafile), (extension .WMF), images vectorielles C'est l'équivalent du format Pict en environnement Windows; capable de dessiner des formes géométriques de façon vectorielles.

Photoshop (extension .PSD), images Bitmap C'est le format natif du logiciel Adobe Photoshop qui est un standard de fait en matière de traitement d'images 2D; de ce fait ce format est de plus répandu et supporté par la plupart des logiciels professionnels d'imagerie du marché. Son principale avantage est de gérer des couches multiples : chaque couche peut être éditée indépendamment des autres. Les fichiers sauvegardés dans ce format sont bien entendu beaucoup plus volumineux qu'avec les autres formats : ce format est pratique tant que des modifications doivent être apportées aux images; il n'est pas adapté au stockage définitif ou au transfert sur réseau.

Image Pac - PCD (Photo CD Kodak) Le format Image Pac (Kodak) est le format de stockage utilisé sur les photos CD (obtenus à partir d'inversibles ou de négatifs 24 x 36). Il en existe une version pro et une version particulier. Dans la version particulier, un photo CD photo peut contenir environ 100 images maximum dans les résolutions 1/16, 1/4, 1/1, 1x4 et 1x16. La version pro ne contient qu'environ 30 images, mais ajoute la résolution 1x64. Le tableau

ci-dessous présente les 6 résolutions des Photos CD, leurs poids et leurs tailles approximatives en fonction de la résolution de sortie (d'impression) choisie :
Type d'image : Résolution Base/16 (vignette) 128 x 192 Base/4 256 x 384 Base 512 x 768 Base x 4 1024 x 1536 Base x16 2048 x 3072 Base x 64 (PRO) 4096 x 6144

Les photos CD sont lisibles sur les lecteurs de CD-ROM reconnaissant la norme XA et sur les lecteurs de CDI (Compact Disk Interactive). Le numériseur (scanner) et le graveur photo CD sont très coûteux et généralement les travaux de création de photo CD sont sous-traités. Les négatifs 35 mm sont scannés à la résolution de 2048 x 3072 pour chacun des 3 canaux (rouge, vert, bleu) : chaque image à ainsi une taille de 18 Mo. Le format Image Pac est un format avec compression sans perte d'information qui permet de passer de 18 Mo à 4,5-6,5 Mo par image.

FlashPix (Kodak), (extension .FPX), image Bitmap. Ce format standard multi-résolutions s'applique à tous les types d'images numériques bitmap. Comme le Photo CD les images sont stockées sous plusieurs résolutions spatiales indépendantes; chaque image étant subdivisée en de multiples zones carrées de 64 x 64 pixels. Ceci permet aux applications de choisir la zone de l'image et la résolution la mieux adaptée pour un traitement particulier. Ceci permet une augmentation de la vitesse de traitement. Ce format supporte également plusieurs modes de compression : JPEG et Single Color. Un fichier au format FlashPix non compressé occupe environ 33% de place en plus sur le disque que le même fichier au format TIFF non compressé (ceci le rend peu intéressant pour du stockage définitif) mais il nécessite moins de RAM pour l'affichage (20% en moins), et il est plus rapide à modifier. Un autre avantage de ce format est la gestion des couleurs selon le mode ICC (International Color Consortium). Malgré ces avantages et le fait qu'il soit supporté par Kodak, Microsoft, Hewlett-Packard, Apple, Canon, Corel, IBM, Intel, Macromedia et Live Picture il tarde à s'imposer et n'est actuellement pas encore très utilisé.

6.2 Formats compressés avec perte d'information

Il faut savoir que l'enregistrement d'une image dans un format compressé avec perte d'information ne doit être réalisé qu'une seule fois, car chaque enregistrement successif provoque un cumul des dégradations qui finissent par devenir très perceptibles. La compression avec perte de donnée doit donc être utilisée avec précaution (et une seule fois) et principalement sur des images finales, dans un but de stockage définitif ou de transfert de photographies sur un réseau par exemple.

En imagerie scientifique, il est fortement conseillé de toujours garder un original des fichiers dans un format non compressé avec une méthode à perte d'information et, de ce fait, on peut pratiquement dire que l'utilisation de formats de compression avec perte d'information doit être proscrite.

JPEG (Joint Photographic Expert Group), (extension .JPG) Ce format, qui est plutôt une méthode de codage, est particulièrement bien adapté aux images 24 bits haute définition; il est reconnu par tous les navigateurs Web. L'algorithme de compression utilisé est un algorithme à perte d'information modulable : plus le taux de compression choisi est important plus le

fichier original est dégradé. Il est remarquable que dans certains cas (images de microscopie confocale), l'application de la compression JPEG se comporte comme un filtre atténuateur de bruit.

Outre la publication d'images sur le Web, ce format peut être utilisé pour des bases de données d'images ou, dans certains cas, du stockage définitif. Ce format n'est généralement pas utilisé pour les images destinées à l'imprimerie et, comme tous les formats de fichiers avec perte de données, on peut considérer que son utilisation n'est pas à recommander dans le domaine scientifique (sauf cas particuliers); il a d'ailleurs été créé plutôt pour les besoins des photographes : c'est le format que l'on retrouve actuellement sur tous les appareils photo numériques. Le taux de compression des fichiers peut atteindre un facteur 20.

6.3 La vidéo numérique

Avec la vidéo numérique, on conjugue deux angles d'attaque :

- La réduction du volume de chaque image, on parle alors de compression spatiale. Les méthodes de compression spatiale concernent en premier lieu les images fixes, elles peuvent être ou non dégradantes.
- La réduction du flux d'informations (en ne codant, pour une image donnée, que ses différences avec la précédente, par exemple), on parle alors de compression temporelle, qui peut, elle aussi, être ou non dégradante.

6.3.1 Les *codecs*

En ce qui concerne la compression temporelle, deux approches sont combinées :

- codage par différence entre des images successives, ce qui risque de conduire à des déceptions sur le volume gagné, en effet, l'image est par nature en mouvement, et deux images successives risquent de présenter beaucoup de différences quand on les regarde à l'échelle du pixel.
- codage par prédiction de mouvement, qui réduit efficacement les volumes, mais introduit souvent une dégradation, surtout si les mouvements sont rapides ou imprévisibles.

La compression temporelle entrave l'accès aux images pour les fonctions d'arrêt, de retour arrière et de recherche rapide, puisque certaines images n'y sont codées qu'en terme de différences les unes par rapport aux autres.

La plupart des algorithmes de compression de l'image vidéo combinent les compressions spatiale et temporelle, essayant de trouver un équilibre pour minimiser les dégradations, qui restent cependant importantes. On s'en serait douté, compte tenu des coefficients qui ont été donnés.

Restituer des images compressées nécessite une décompression suffisamment rapide pour assurer un flux correct à l'affichage. L'opération a donc deux versants, encodage/décodage, qui, réunis, constituent un *codec*. L'encodage met en œuvre des

algorithmes de compression effectuant des calculs complexes, soit en temps différé avec des logiciels et des matériels puissants et rapides. (MPEG 1, en studio haut de gamme), soit en temps réel avec l'adjonction de matériels spécialisés (cartes additionnelles pour les encodages, qu'ils soient en MPEG 1, d'extension AVI ou sous QuickTime). La restitution utilise l'algorithme de décompression inverse de celui employé pour la compression. En principe, cette opération devrait garantir un débit de 25 ou 30 images par seconde, ce qui n'est en général pas le cas.

Note: Le logiciel *ImageJ* utilise le codec RGB uncompressed lorsqu'on sauve en AVI, il faut alors utiliser un programme externe pour changer de codec alors qu'il laisse le choix du codec lorsqu'on sauve au format *QuickTime* (voir ci-après).

6.3.2 Les formats

QuickTime

(extension .MOV), Format de référence sous Mac OS; fonctionne aussi sous Windows (toutes versions). Ce format est très pratique pour l'échange de données vidéo entre les plateformes Mac et PC.

AVI

(extension .AVI), Format de référence sous toutes les déclinaisons de Windows (logiciels : Vidéo for Windows et Windows Media Player).

MPEG I

Ce format convient bien aux médias véhiculant des débits peu élevés (environ 1,5 Mbit par seconde); il est bien adapté aux lecteurs de CD-ROM (taux de transfert minimum de 150 Ko/s soit 1,2 Mbits/s). L'encodage des fichiers se fait au moyen d'une carte spécifique; le décodage peut se faire de manière logicielle (processeur rapide) ou matérielle. La résolution de base est de 352 x 240 (format SIF; en PAL); 352 x 288 (format SIF; en NTSC).

MPEG II

Ce format a été défini pour les applications de télédiffusion (télévision numérique à haute définition) et les disques DVD; il supporte des débits allant de 4 Mbits/s à 20 Mbits/s. La compression se fait par une carte spécifique et la décompression peut être soit matérielle (conseillé) soit logicielle (Pentium II 400 minimum). Les résolutions gérées sont de 352 x 576 (format HD1 = Half D-1); 720 x 576 (format FD1 = Full D-1).

Le format MPEG II est un des plus répandu actuellement et il offre des bons taux de compression.

Filtrage spatial

Christophe CHAMOT

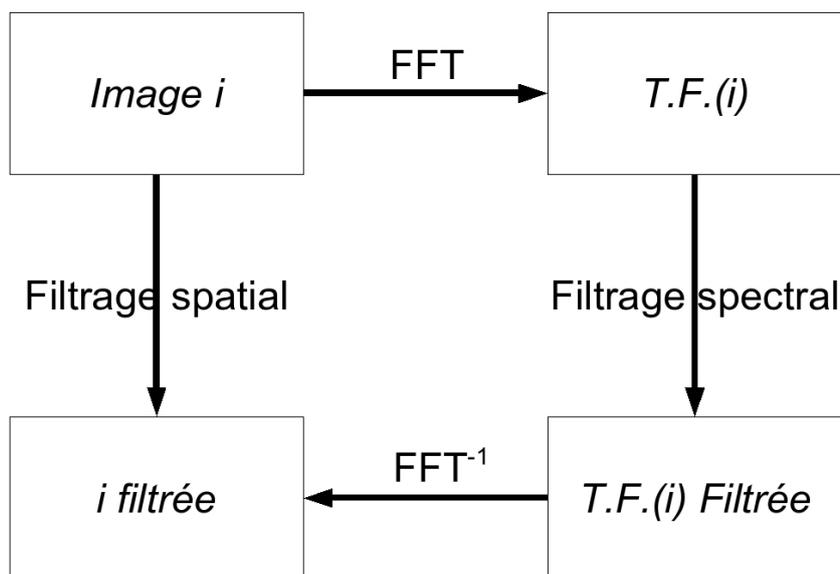
14 Mars 2007

Table des matières

1	Convolution discrète	3
1.1	Compréhension intuitive	3
1.2	Un peu de maths	3
2	Description des filtres	4
2.1	Filtres linéaires	4
2.1.1	Filtre moyennneur	4
2.1.2	Filtre gaussien	5
2.2	Filtres dérivatifs	6
2.2.1	Approche gradient, filtre de Roberts	6
2.2.2	Filtre de Prewitt et de Sobel	6
2.2.3	Laplacien et dérivé de gaussienne	7
2.3	Filtres non-linéaires	9
2.3.1	Filtres de rang	9
2.4	Filtres adaptatifs	9
2.4.1	Moyenne seuillée	9
2.4.2	Moyenne pondérée	9
2.4.3	Kuwahara	10
3	Étude comparative	11

Introduction

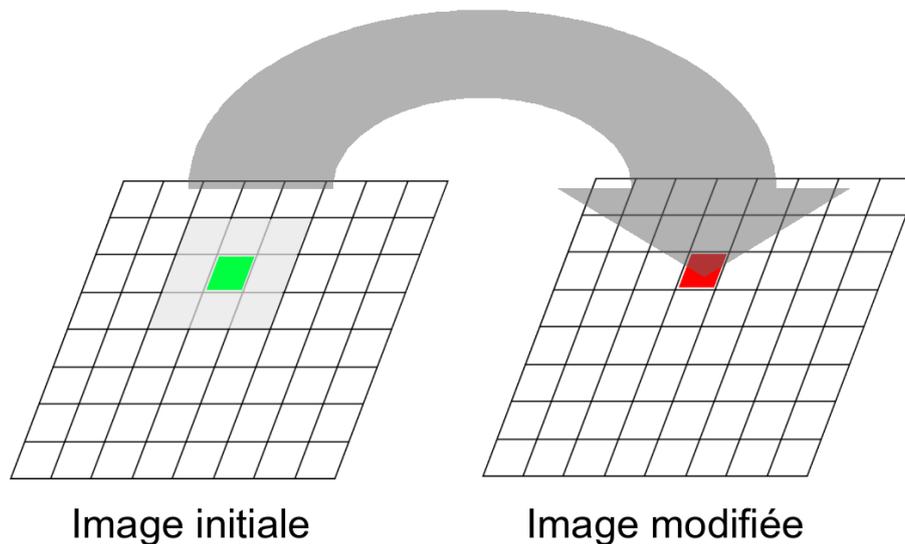
Le filtrage peut être abordé de deux façons, un aspect spatial et un spectral. Ces représentations sont parfois équivalentes d'ailleurs, l'aspect spatial peut nous sembler plus « naturel » car nous sommes des êtres essentiellement visuels.



F.F.T. = Fast Fourier Transform

1 Convolution discrète

1.1 Compréhension intuitive



La nouvelle valeur -en rouge- du pixel vert, est le résultat d'une opération sur tous ses voisins (gris).

1.2 Un peu de maths

La convolution discrète est un outil permettant la construction de filtres linéaires ou de filtres de déplacements invariants. Pour un filtre de largeur D , elle est définie mathématiquement par la relation :

$$(I \otimes F)(x, y) = \sum_{i=-(D-1)/2}^{(D-1)/2} \sum_{j=-(D-1)/2}^{(D-1)/2} I(x+i, y+j) \times F(i, j)$$

Si I est l'image, F est souvent appelé filtre, noyau, masque de convolution, ... En pratique, la convolution se fera par une sommation de multiplications.

2 Description des filtres

2.1 Filtres linéaires

L'image filtrée (I_f) sera le résultat de la convolution de l'image brute (I_b) par un filtre dont la taille —on parle parfois de *rayon* ou de *largeur* du filtre— et les coefficients varient. Plus la taille du masque sera grande, plus l'effet du filtrage sera fort.

2.1.1 Filtre moyenneur

Exemple d'un filtre moyenneur de largeur 3 :

$$I_f = I_b \otimes \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Si I_b est de la forme :

$$\begin{pmatrix} A & B & C & D & E \\ F & G & H & I & J \\ K & L & M & N & O \\ P & Q & R & S & T \\ U & V & W & X & Y \end{pmatrix}$$

La nouvelle valeur de G sera donc : $1/9 \times (A + B + C + F + G + H + K + L + M)$

Le coefficient $1/9$ est là pour conserver la valeur moyenne de l'image, pour deux raisons ; la première est la conservation d'une luminosité globale, la seconde pour que les données puissent « tenir » dans l'image filtrée, sans générer d'erreur de dépassement de capacité liée à la dynamique utilisée. En clair, si une image ne peut contenir que 256 niveaux d'intensité, difficile d'y faire rentrer une somme de 9 pixels voisins sans dépasser cette valeur...

Une question fréquemment posée est le devenir des pixels du bord, prenons K par exemple. On pourrait faire la moyenne des voisins présents ($(1/5) \times (F + G + L + P + Q)$) mais ce n'est pas très satisfaisant pour au moins deux raisons. La première est que ce pixel n'a pas de raison de subir un traitement différent de ses camarades car si il marque la fin de l'image, il ne marque pas la fin de l'objet observé ; en fait, on induit un biais qui est la surestimation de la contribution des pixels du bords. La seconde est le rallongement du temps de calcul car il faudrait non seulement traiter tous les pixels de bords différemment mais en plus prévoir la détection d'un pixel de bord ... La solution la plus simple et la plus utilisée est le repliement.

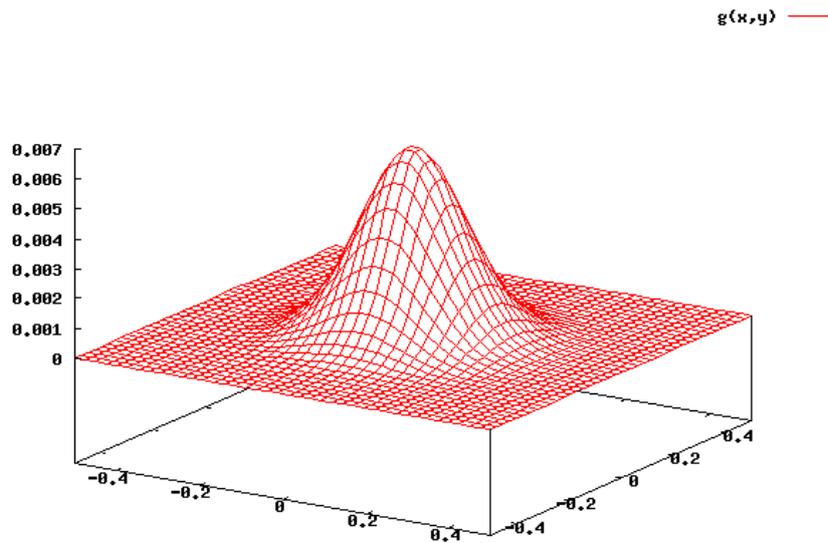
On considère que le bord de l'image est un miroir et on a des valeurs pour calculer notre nouveau K (en l'occurrence $G, F, G, L, K, L, Q, P, Q$).

2.1.2 Filtre gaussien

Évidemment, les coefficients du filtre ne sont pas toujours 1, on peut vouloir accorder un poids plus important au pixel en cours qu'à ses voisins par exemple. Une façon très simple de faire est de remplir le masque de convolution avec les coefficients d'une gaussienne en 2D :

$$g(x, y) = \frac{1}{2\pi\sigma^2} \cdot e\left(-\frac{x^2+y^2}{2\sigma^2}\right)$$

On peut très facilement jouer sur la « force » du filtre avec le paramètre σ (la largeur de la gaussienne en quelque sorte).



Gaussienne 2D $\sigma = 3$

Exemple de coefficients pour un noyau de rayon 5 :

0.114	0.249	0.458	0.707	0.917	1.000	0.917	0.707	0.458	0.249	0.114
-------	-------	-------	-------	-------	-------	-------	-------	-------	-------	-------

Ne sont représentées que les valeurs de la ligne centrale, tous les autres coefficients sont construits par symétrie circulaire.

2.2 Filtres dérivatifs

La principale application de ces filtres est la détection (ou le renforcement) des contours dans une image.

2.2.1 Approche gradient, filtre de Roberts

Le gradient est l'opérateur de base pour trouver un contour. C'est la première dérivée de l'image qui s'exprime :

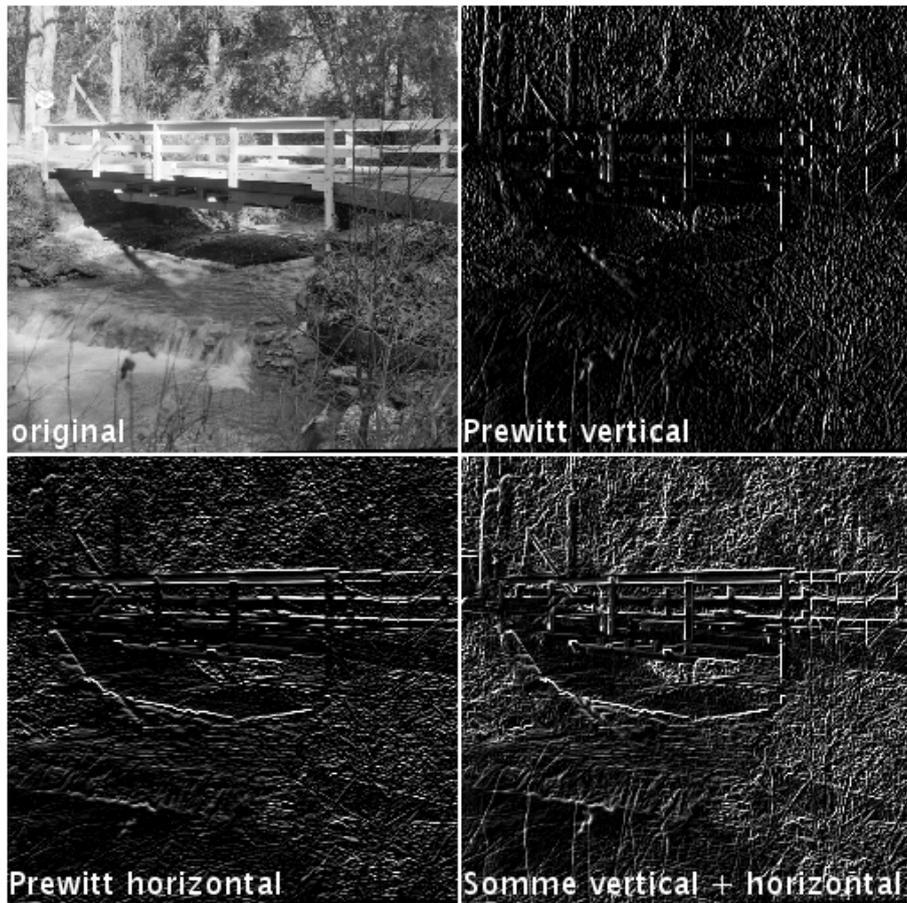
$$|\nabla f| = \left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2$$

Roberts en donne la première approximation discrète en 1965, à l'aide de deux masques de convolution qui définissent la dérivée dans deux directions orthogonales $\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$ et $\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, G_x et G_y . On peut alors extraire deux paramètres intéressants, la norme du gradient : $|G| = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y|$ et sa direction : $\theta = \arctan G_x/G_y - 3\pi/4$

2.2.2 Filtre de Prewitt et de Sobel

Ils se décomposent aussi en deux filtres, pour détecter les bords horizontaux et verticaux de l'image.

$$\begin{aligned} \text{Prewitt : } & \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \text{ et } \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \\ \text{Sobel : } & \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \text{ et } \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \end{aligned}$$



2.2.3 Laplacien et dérivé de gaussienne

On peut aussi trouver les contours par la dérivée seconde de l'image. Si ils correspondaient aux maxima de la dérivée première, ils correspondront aux passages par zéro de la dérivée seconde. On utilise le Laplacien comme opérateur :

$$\nabla^2 I_b = \frac{\partial I_b}{\partial x^2} + \frac{\partial I_b}{\partial y^2}$$

Encore une fois, plusieurs approximations discrètes existent :

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -1 & 1 \\ 0 & 1 & 0 \end{bmatrix} \text{ ou } \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \text{ ou } \dots$$

L'avantage est de n'avoir ici qu'un seul filtre, donc une seule convolution à faire. De plus, le filtre est symétrique en rotation, ce qui n'était pas le cas des filtres précédents de détection de contours.



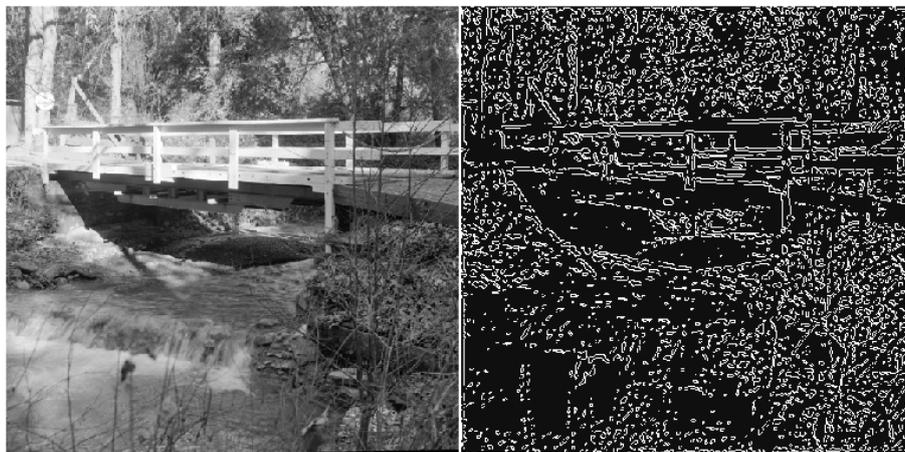
Comparaison Sobel et Laplacien

L'inconvénient majeur est la forte sensibilité au bruit. On va donc lisser l'image avec une gaussienne avant de détecter les contours. Cette opération se fait en une fois, car on pré-convolue la gaussienne avec le Laplacien :

$$\nabla^2(I_b \otimes G) = I_b \otimes (\nabla^2 G) \text{ avec } G(x, y) = \frac{1}{2\pi\sigma^2} \cdot e^{\left(-\frac{x^2+y^2}{2\sigma^2}\right)}$$

$$\Rightarrow LoG(x, y) = -\frac{1}{\pi\sigma^4} \left[1 - \frac{x^2 + y^2}{2\sigma^2}\right] e^{\frac{x^2+y^2}{2\sigma^2}}$$

Cette fonction a la forme d'un sombrero inversé, on retrouve donc ce filtre sous l'appellation anglo-saxonne « mexican hat »



Laplacien de gaussienne $\sigma = 3$

avec $K = \frac{(Var-\sigma)}{Var}$

2.4.3 Kuwahara

On définit ici quatre régions, A, B, C et D de la façon suivante :

A	A	AB	B	B
A	A	AB	B	B
AC	AC	ABCD	BD	BD
C	C	CD	D	D
C	C	CD	D	D

On calculera ensuite la moyenne et la variance pour chaque zone. La nouvelle valeur du pixel $ABCD$ sera la moyenne de la zone qui a la variance la plus faible.

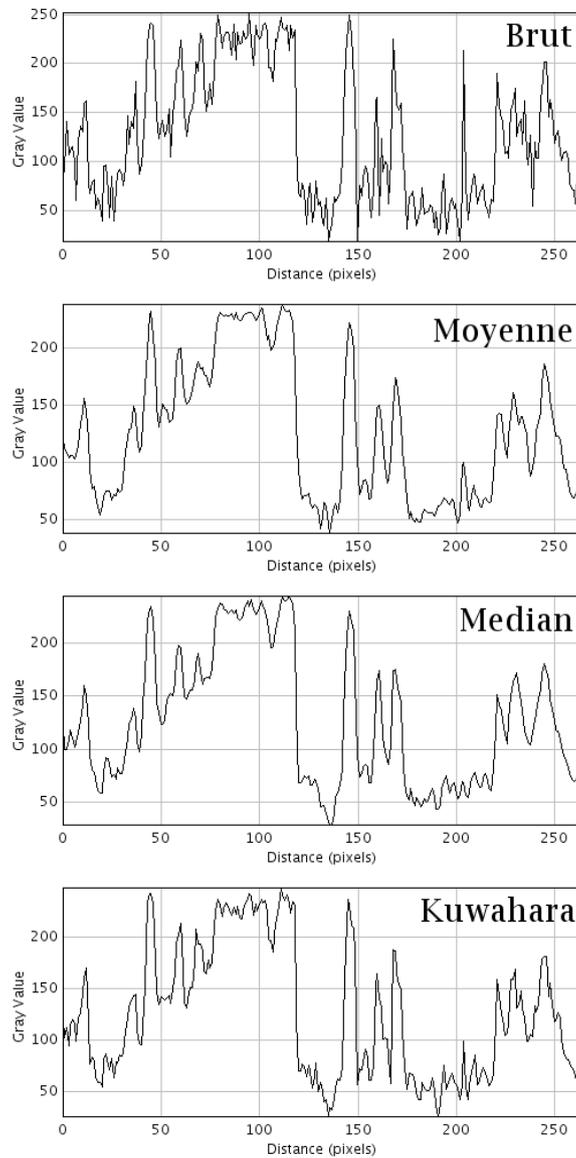
3 Étude comparative

Nous allons utiliser une image bruitée et appliquer trois filtres différents et observer un profil d'intensité.



Comparaison des effets des filtres étudiés

Voilà ci-après les profils d'intensité selon la même ligne quelconque à travers les images précédentes, dans l'ordre, c'est-à-dire image brute, moyenne, médiane et Kuwahara.



On voit très bien l'effet de ces filtres sur ces profils. Le moyennneur diminue les transitions très rapides mais ne les annule pas. Le médian coupe vraiment les transitions rapides mais « floute » aussi les plus lentes. Le filtre adaptatif présente des caractéristiques du moyennneur mais respecte bien mieux les transitions.

Yves TOURNEUR

Filtrage fréquentiel

Analyse de Fourier et filtrage fréquentiel

- ▷ Composantes fréquentielles 1D (profil)
- ▷ Coefficients de Fourier
- ▷ Composantes fréquentielles 2D (image)
- ▷ Applications de filtrages
- ▷ Compléments : effet stroboscope, Jpeg

yves.tourneur@univ-lyon1.fr

2

Analyse fréquentielle en 1D

- ▷ Analyse d'un profil
- ▷ Principe de la décomposition fréquentielle
 - ▷ Analogie acoustique
 - ▷ Spectre optique
- ▷ Principe de l'analyse de Fourier
 - ▷ Cas d'un signal périodique
 - ▷ Reconstitution du signal à partir des composantes harmoniques

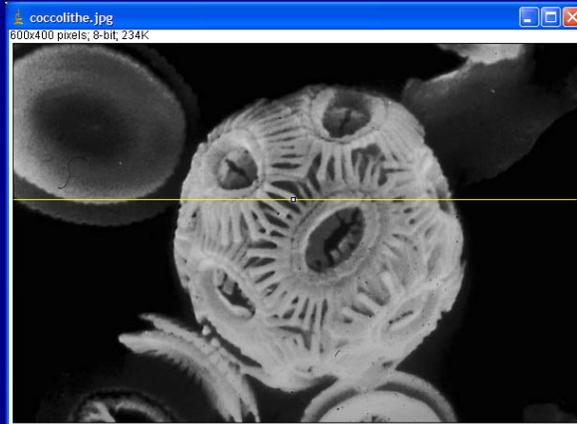
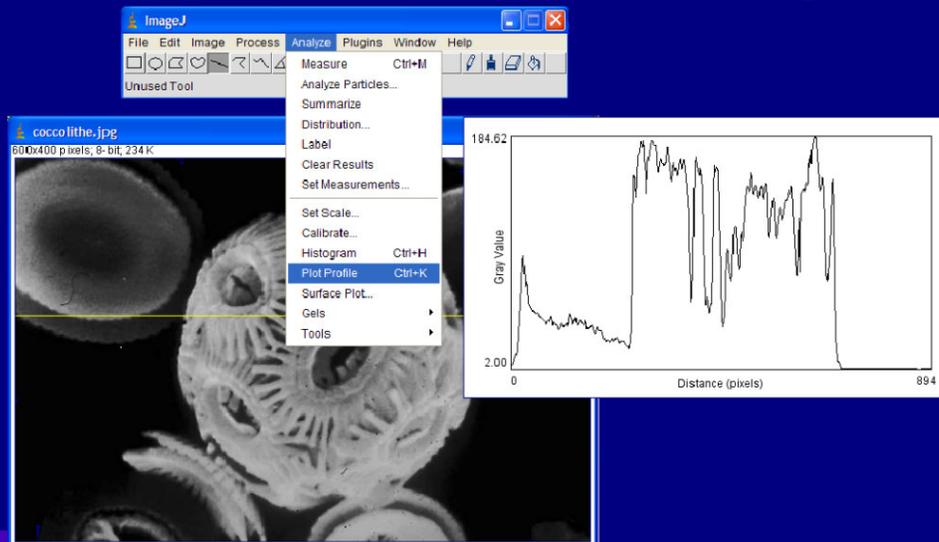
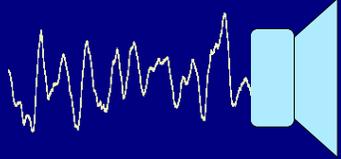


Image en niveaux de gris

Analyse d'une ligne de l'image



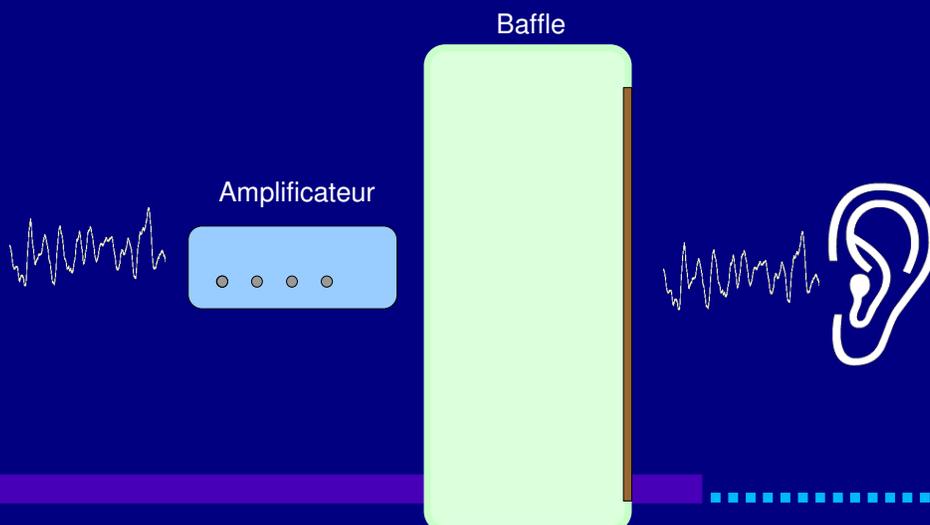
Transformation en signal audible

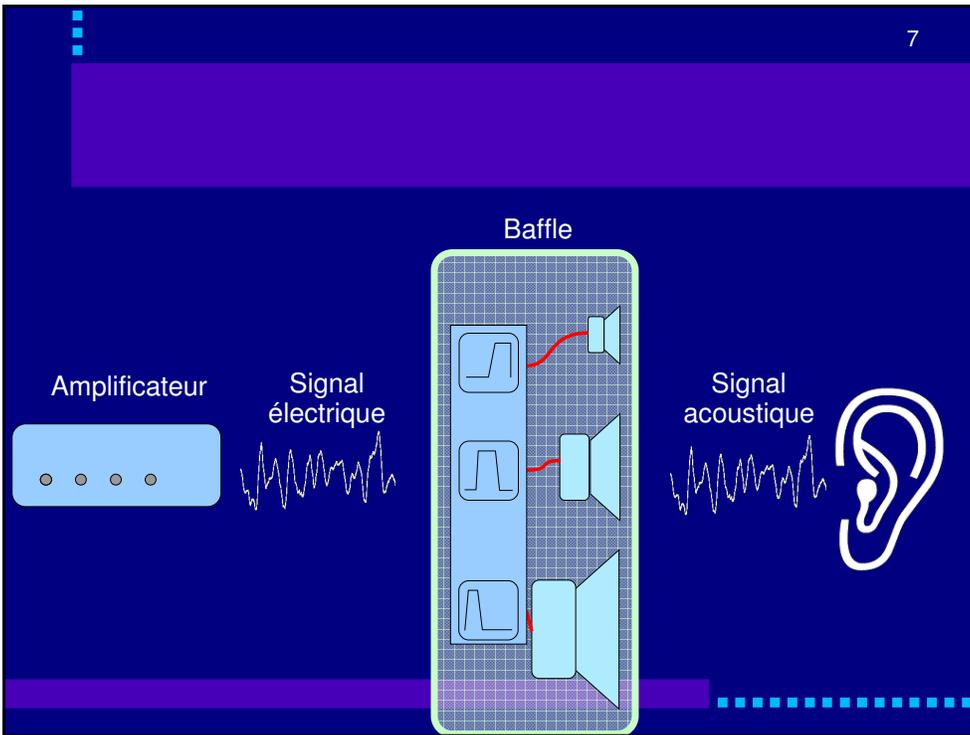


Transformation en
signal électrique
de la ligne, jouée en
boucle

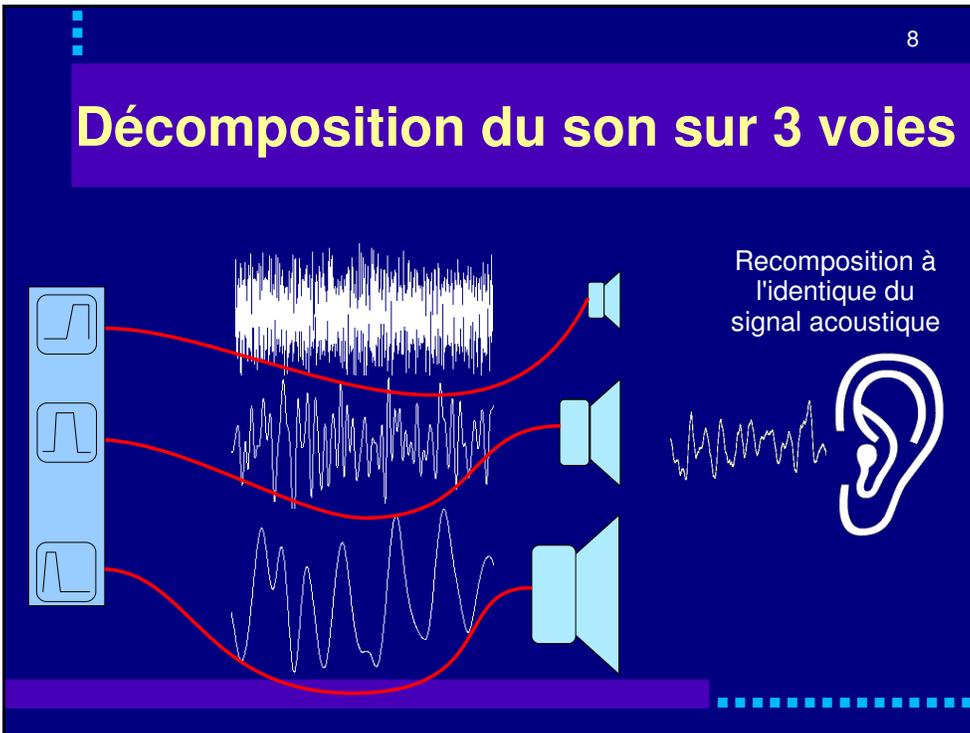


Autre reproduction du son

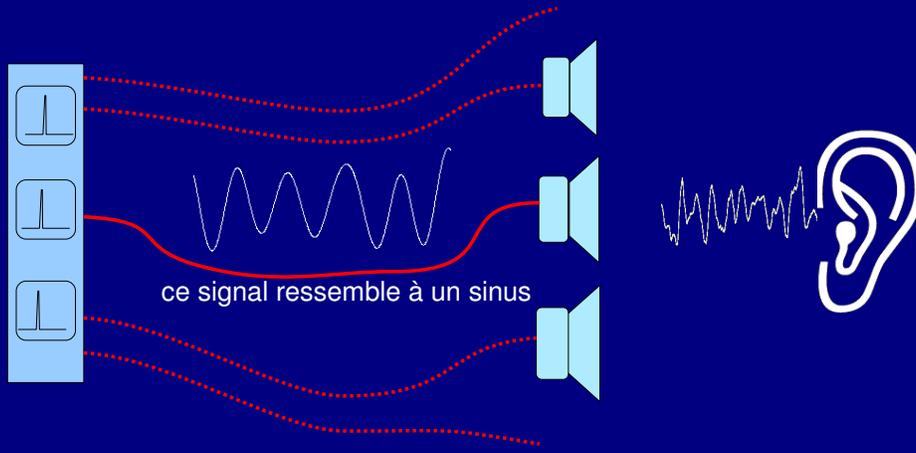




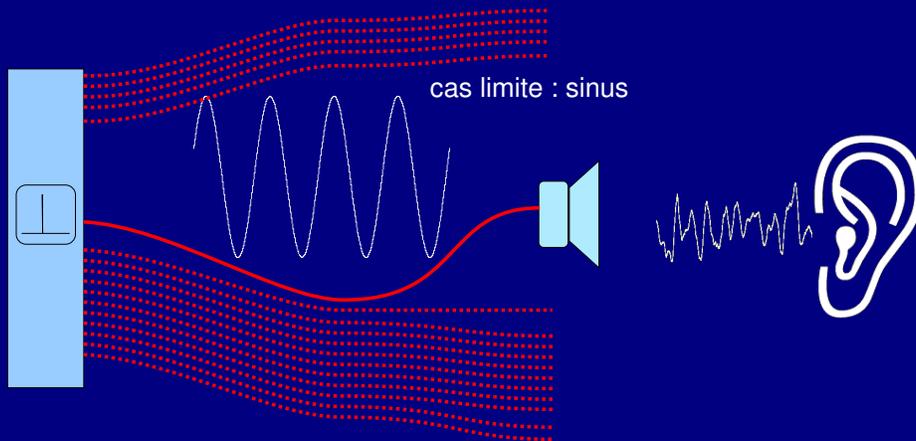
Décomposition du son sur 3 voies



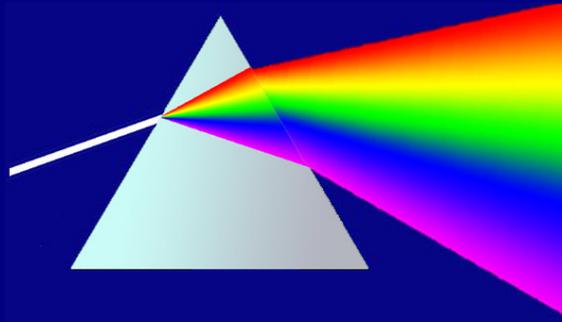
Décomposition plus fine



Une seule fréquence par voie



Analogie : spectre lumineux

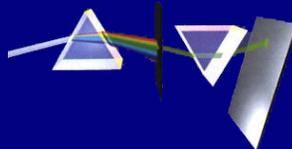


Pour Aristote, le prisme déforme la lumière

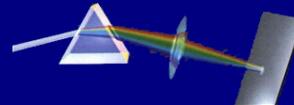
Rouge
Orangé
Jaune
Vert
Bleu
Violet

Décomposition - Recomposition

Newton a montré que le prisme ne modifie pas la lumière blanche, mais sépare des éléments qui peuvent se recomposer pour redonner de la lumière blanche.

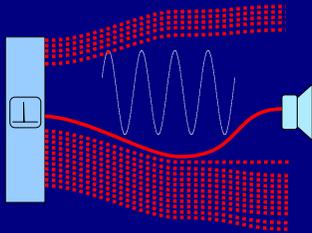


Le prisme ne modifie pas une couleur simple



Les couleurs peuvent être recomposées pour redonner de la lumière blanche

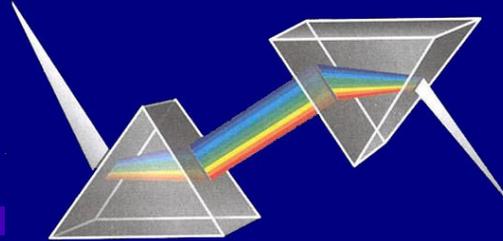
Décomposition spectrale



Décomposition spectrale

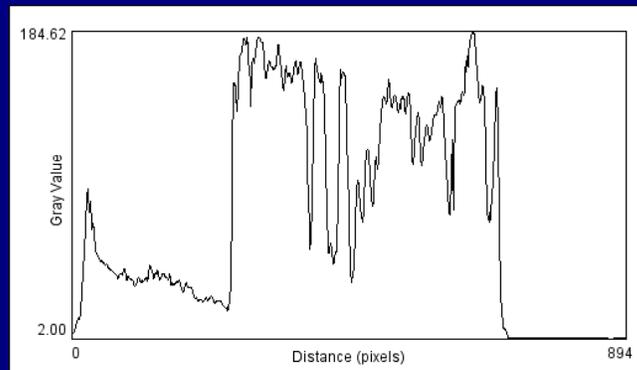


Recomposition du signal d'origine



Décomposition du profil en sinus

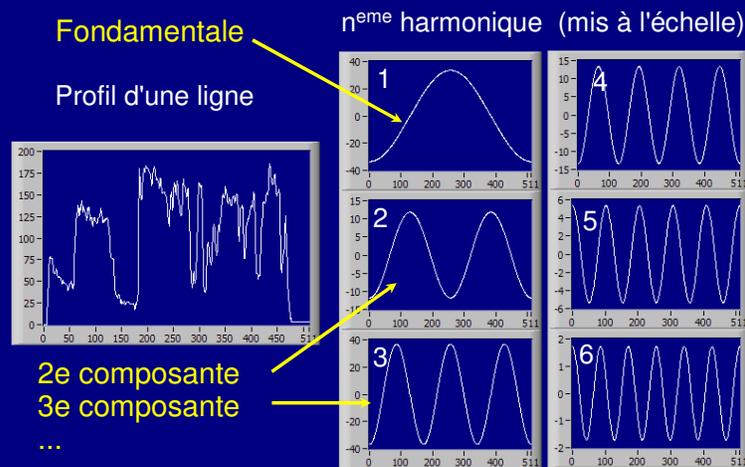
↳ Décomposition du profil joué en boucle



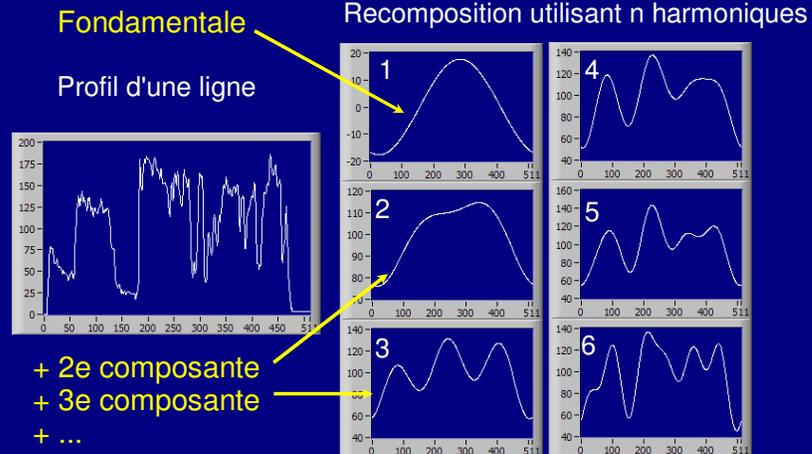
Cas d'un signal périodique

- ▷ Si le signal est périodisé, les fréquences ne sont pas quelconques
- ▷ Leur fréquence est multiple d'une fréquence "fondamentale".
- ▷ La période de la fondamentale est la durée du signal

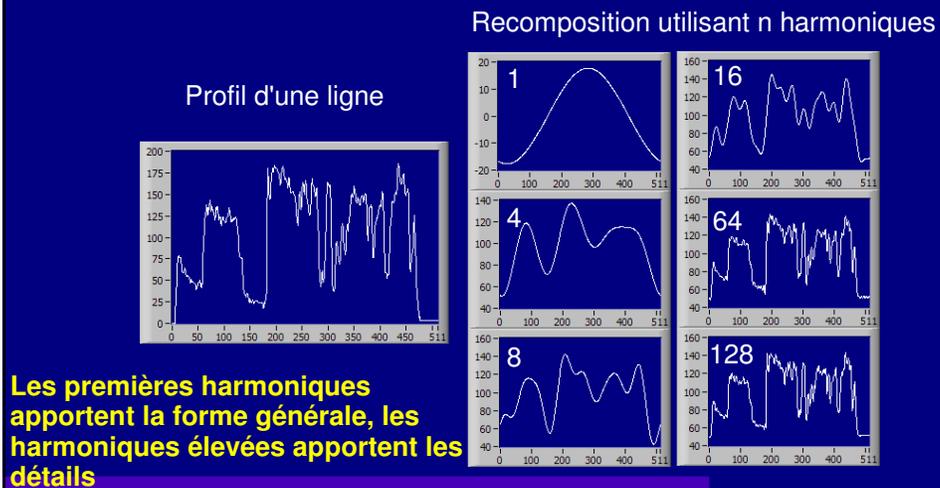
Décomposition harmonique



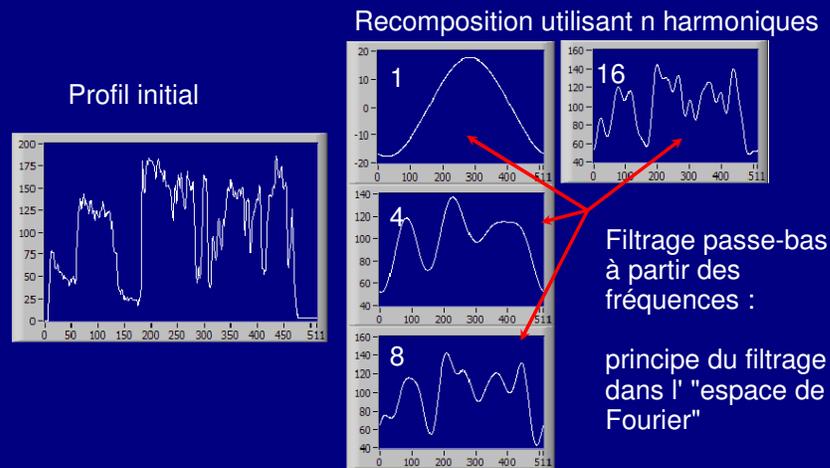
Recomposition du profil



Recomposition du profil



Utilisation pour le lissage



Différents filtrages

- ⇒ On ne reconstitue le signal qu'avec les harmoniques de rang faible : on supprime les composantes rapides : **Filtre passe-bas**
- ⇒ On peut ne mettre que les harmoniques de rang élevé : on supprime alors les composantes lentes : **Filtre passe-haut**
- ⇒ On peut ne mettre que les composantes intermédiaires : **Filtre passe-bande**

Principe du calcul :

Les coefficients de Fourier



Principe de l'analyse de Fourier

- ▷ Réalité physique : tout signal physique peut être décomposé en somme d'éléments simples : des sinusoïdes. La somme de ces éléments reconstitue le signal d'origine
- ▷ L'analyse de Fourier consiste à établir les méthodes pour obtenir ces composantes par calcul (moindre carrés).

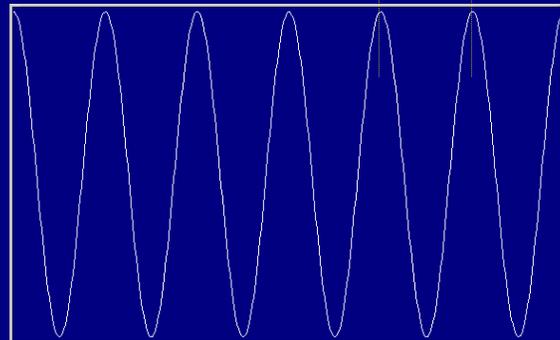
Rappels sur les fonctions sinus :

fréquence f
(en m^{-1}) (en Hz) :
nombre de pics par
mètre seconde

relation entre
ces valeurs :

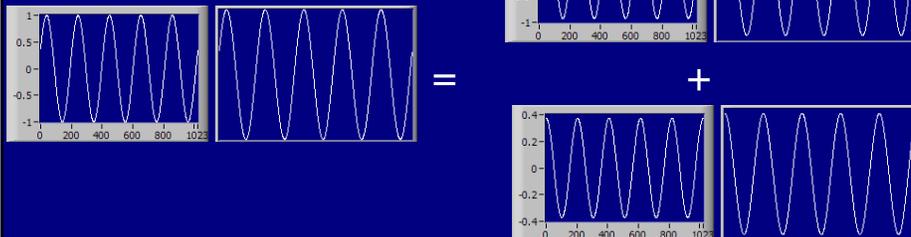
$$T = 1/f$$

période T (en m) (en s) :
distance (temps) entre deux pics



Décomposition en sinus+cosinus

Une fonction sinusoïdale
de phase quelconque est
la somme d'un sinus et
d'un cosinus



$$\sin(t+\phi) = \sin(t).\cos(\phi) + \cos(t).\sin(\phi)$$

La décomposition spectrale :

- ▷ Fournit pour chaque composante :
- ▷ Soit une Amplitude et une Phase

$$\sin(x + \phi) = \sin(x) \cdot \cos(\phi) + \cos(x) \cdot \sin(\phi)$$

- ▷ Soit une partie sinus et une partie cosinus.
C'est la forme utilisée en pratique

Fourier et moindres carrés :

- ▷ Pour une fonction $f(x)$ que l'on veut ajuster avec une fonction $g(x)$. On cherche un coefficient λ tel que $f \sim \lambda g$
- ▷ On cherche à minimiser la somme des différences $(f - \lambda g)^2$ pour toutes les valeurs de x . $\sum (f - \lambda g)^2 = \sum [f^2 - 2 f \lambda g + \lambda^2 g^2]$
- ▷ minimum pour $\lambda = \sum fg / \sum g^2$
- ▷ On calcule $\sum g^2 = 1/a$, il reste donc $\lambda = a \sum fg$

$$\lambda = a \sum f(x) \sin(x)$$

Calcul rapide : la FFT

- ▷ On calcule pour chaque ligne le meilleur ajustement d'une fonction sinus de fréquence donnée. Directement, ce calcul est très long.
- ▷ Méthode de calcul rapide de Cooley et Tuckey, 1965 : la Fast Fourier Transform (FFT). Utilise un nombre de points du type 2^n (256, 512...)
- ▷ Algorithme préconisé par Gauss (1805) ?

Composantes fréquentielles en 2D

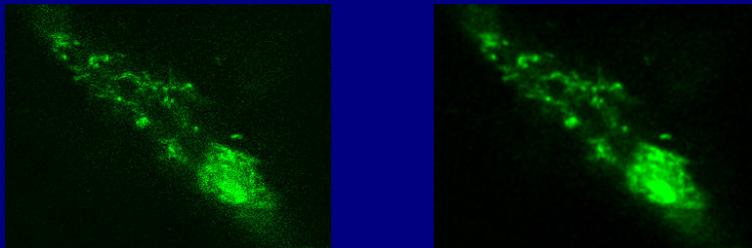
- ▷ Reconstruction 2D à partir des composantes
 - Filtrage passe-bas
 - Filtrage passe-haut
- ▷ Décomposition fréquentielle 2D
 - Représentation du spectre
 - Images-type
 - Filtrage passe-bas, passe-haut, passe-bande
- ▷ Autres applications
 - Suppression, accentuation de raies
 - Effet stroboscope
 - Format Jpeg

Reconstitution d' une image



- ▷ En augmentant le nombre de composantes spectrales, on retrouve les détails
- ▷ Utilisable pour supprimer les détails inutiles

Filtrage passe-bas



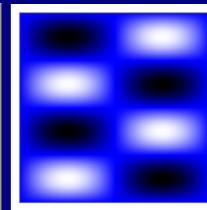
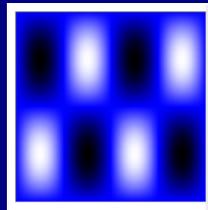
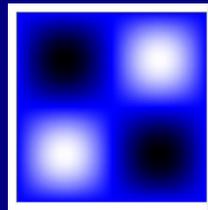
- ▷ Cette image de microscopie confocale contient des détails inutiles : pixels de 'bruit'.
- ▷ Le filtrage les élimine

Filtrage passe-haut



Décomposition fréquentielle 2D

- ▷ L'image est composée en ondes horizontales et verticales
- ▷ On cherche l'ensemble de ces composantes
- ▷ Il est peu utile de connaître la valeur des composantes.

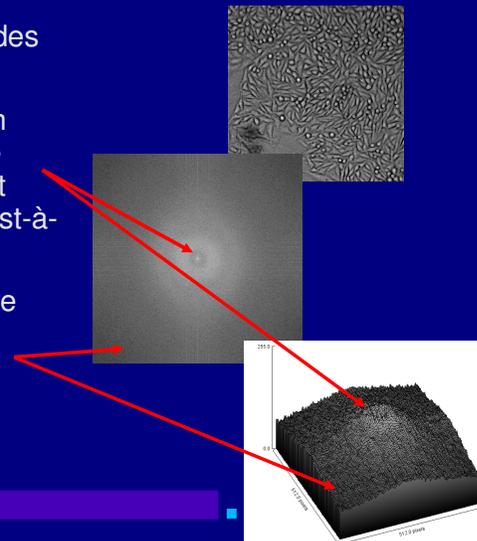


Analyse spectrale d'une image

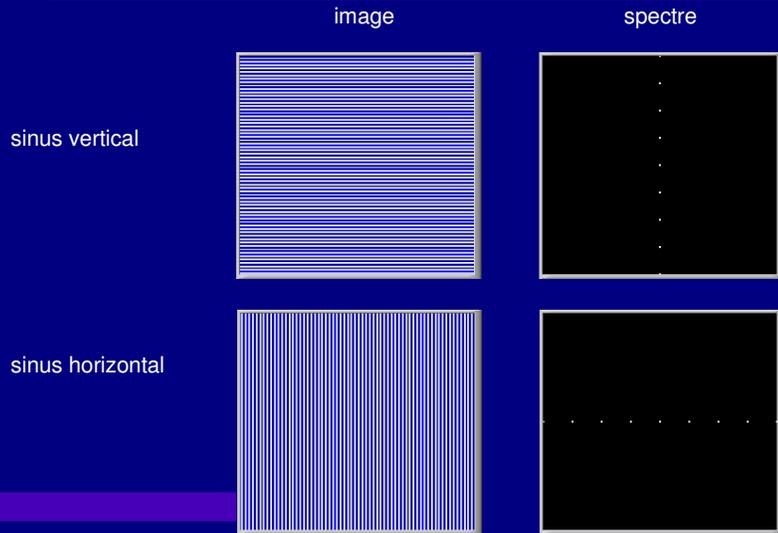
- ▷ On peut décomposer l'image en lignes et calculer les composantes pour chaque ligne.
- ▷ Pour chaque composante harmonique, on décompose l'image en colonnes et on calcule les composantes pour chaque colonne.

Représentation du spectre

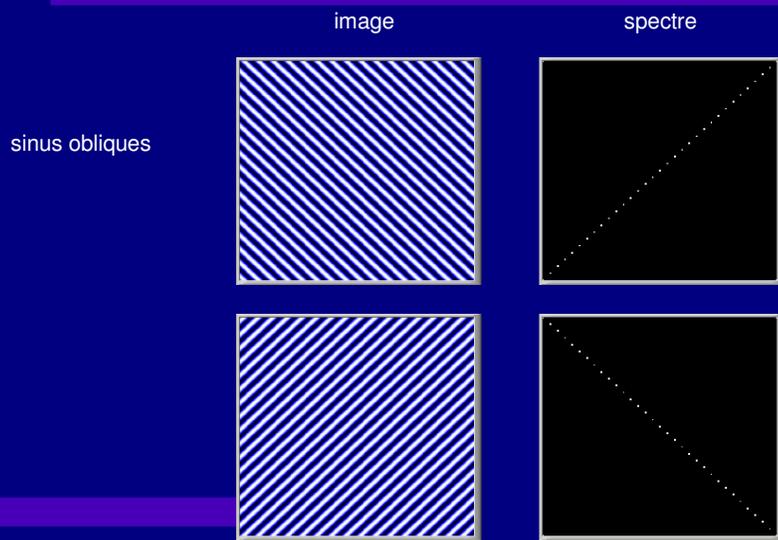
- ▷ Le spectre est la représentation des composantes spectrales
- ▷ Par convention, on représente en général le spectre par une image d'amplitude dans laquelle le point central est la fréquence (0,0), c'est-à-dire la valeur continue
- ▷ Les points plus éloignés du centre représentent les fréquences plus élevées



Exemples de spectres



Exemples de spectres



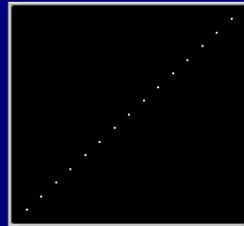
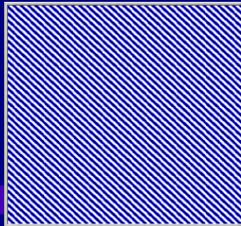
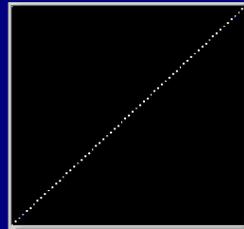
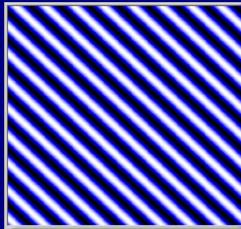
Effet de la fréquence :

image

spectre

raies plus rapprochées :

points du spectre plus éloignés (fréquences plus grandes)

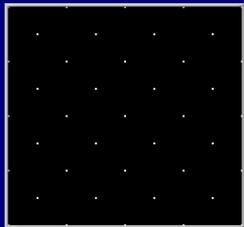
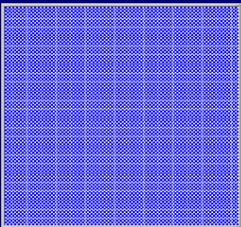


Représentation du spectre

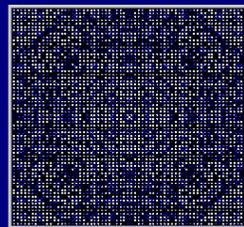
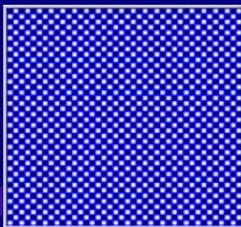
image

spectre

fréquence élevée



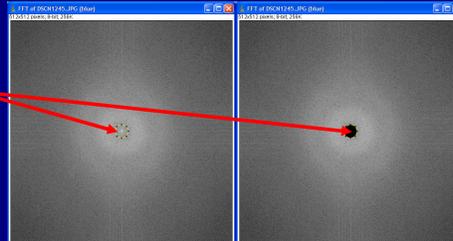
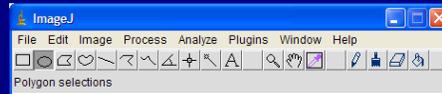
fréquence basse



Filtrage passe-haut en pratique

FFT

Cercle
dessiné
pour
supprimer
les basses
fréquences

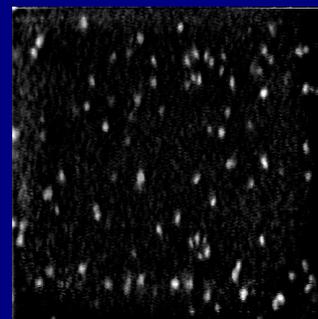
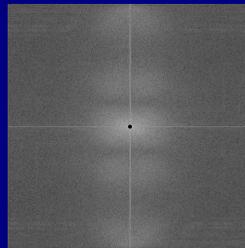
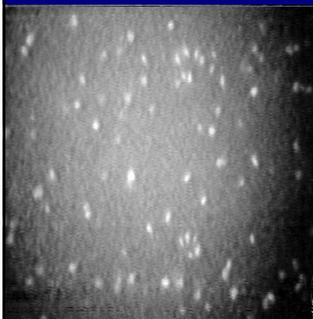


Filtrage passe-haut

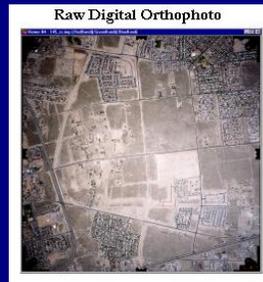
Image originale

Plan de fréquences

Image filtrée



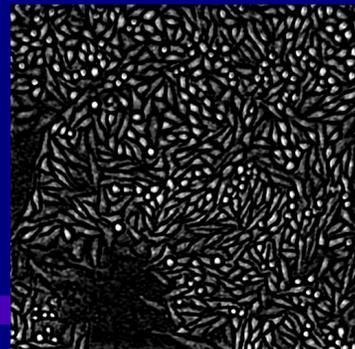
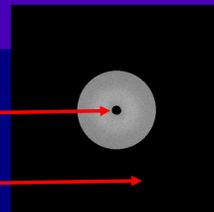
Filtrage passe-haut



- ▷ Cette vue aérienne n'a pas d'éclairage uniforme
- ▷ La suppression des basses fréquences permet de ne conserver que les informations de détail.

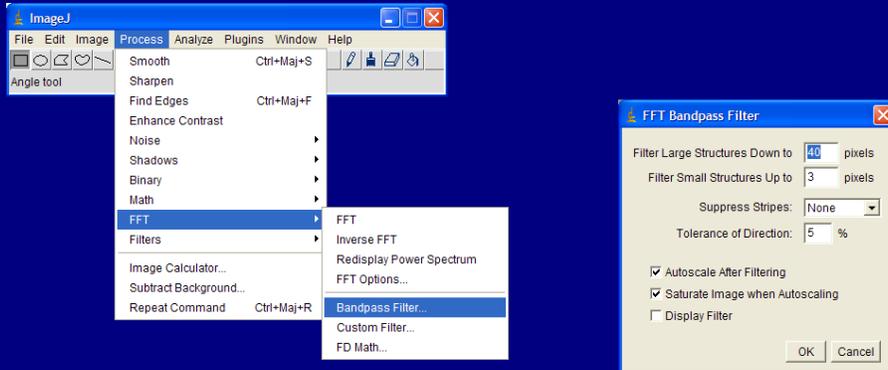
Filtre passe-bande

- ▷ On peut supprimer les fréquences basses et les fréquences hautes.
- ▷ Le résultat est similaire au résultat précédent, mais avec des images plus 'lisses' (suppression des hautes fréquences)
- ▷ Le résultat est obtenu par essai-erreur

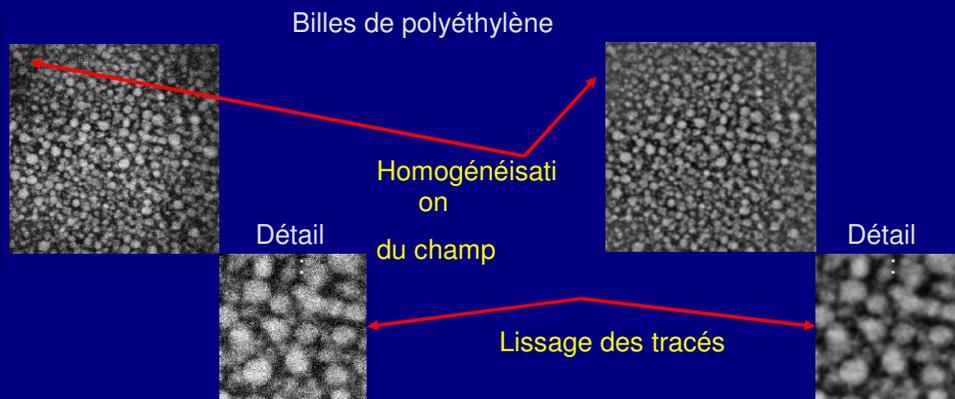


selon la taille des objets que l'on cherche à souligner

➤ selon la taille des objets que l'on cherche à souligner



Exemple :



Autres types d'applications

- ▷ Supprimer des raies
- ▷ Souligner des périodicités

- ▷ Effet stroboscope (repliement de spectre)
- ▷ Format Jpeg

Autres types d'applications

- ▷ Supprimer des raies

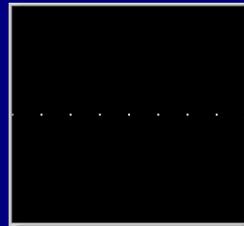
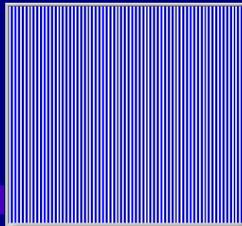
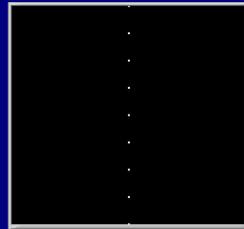
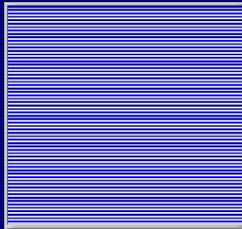
- ▷ Souligner des périodicités

Suppression de raies :

image

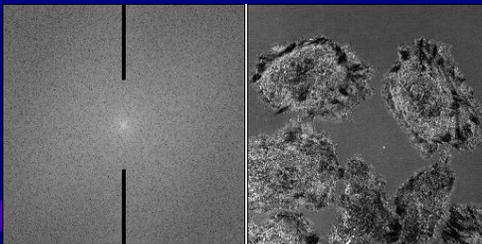
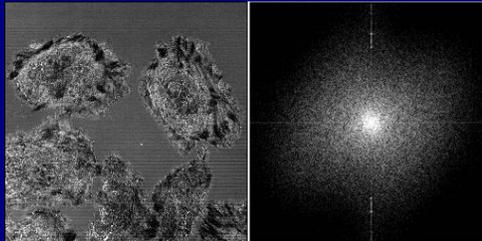
spectre

Rappel :



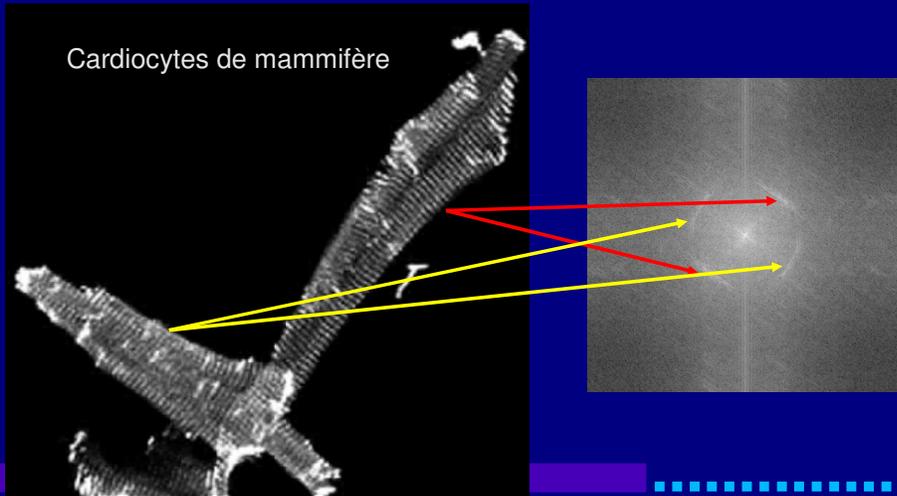
Suppression de raies

↳ Suppression de raies horizontales

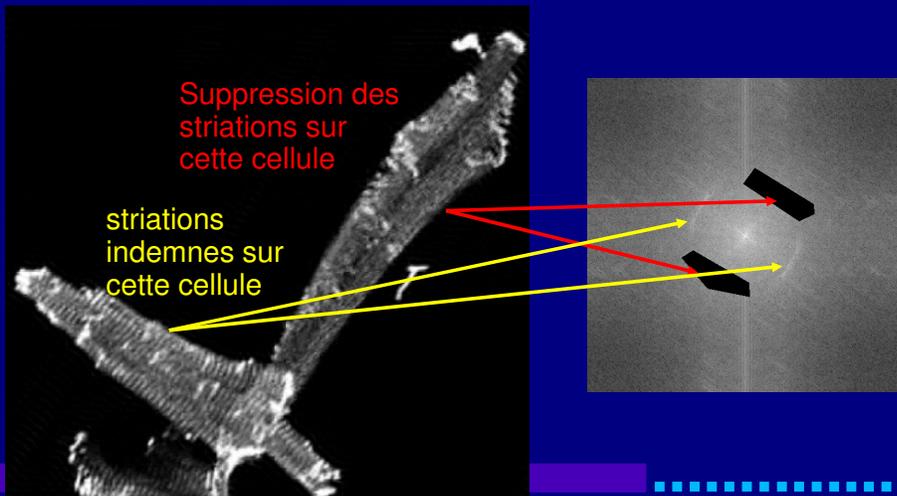


(documentation imageJ)

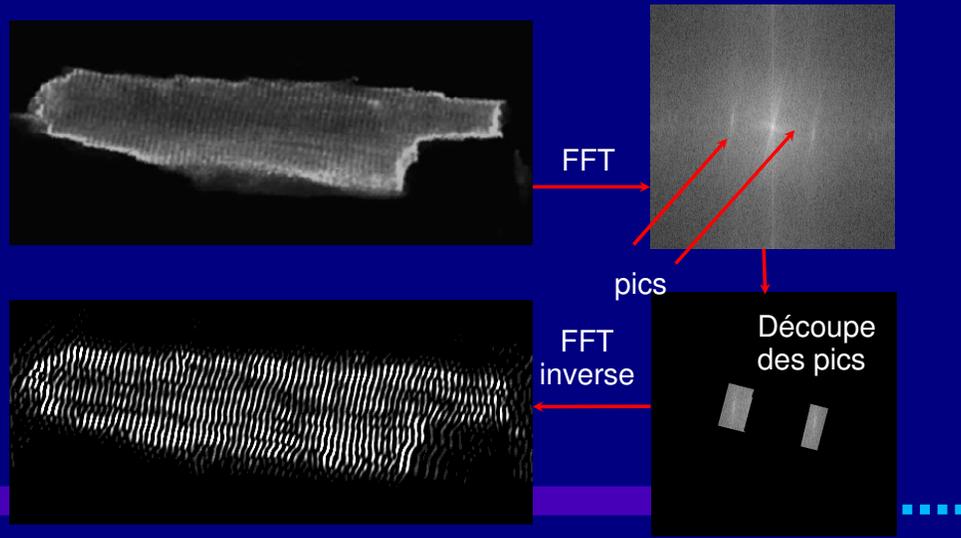
Manipulation d'image périodique



Vérification



Soulignement de la périodicité



En conclusion

- ▷ La suppression des basses fréquences souligne les détails (contours...) et supprime les inhomogénéités de lumière
- ▷ La suppression des hautes fréquences supprime les détails
- ▷ La suppression de verticale centrée supprime les raies horizontales
- ▷ Efficace, mais... pas toujours aisé à prédire.

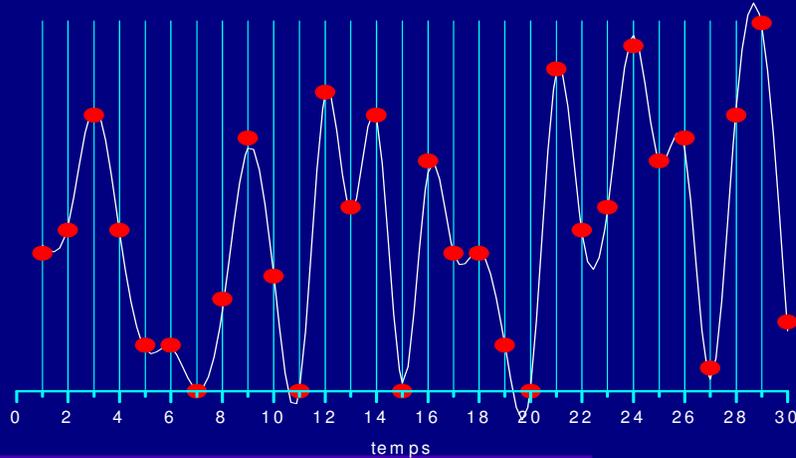
Compléments : l'effet stroboscope

Ou repliement de spectre (Aliasing effect)

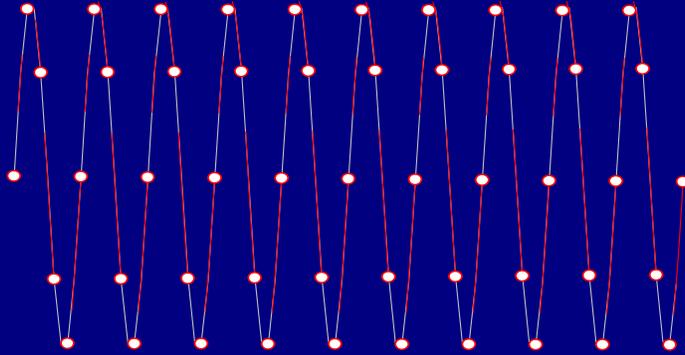
- ▷ Principe sur une ligne
- ▷ Effets sur des images

Échantillonnage

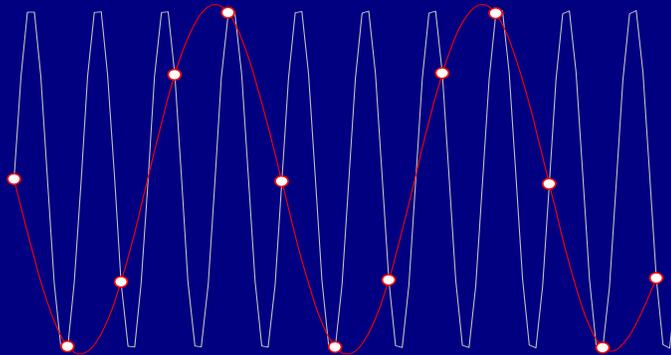
Niveau



Échantillonnage correct



échantillonnage insuffisant : effet de stroboscope.



Equivalent de l'effet stroboscope sur une image



Image scannée 150 ppp



Image scannée 100 ppp

Apparition de réseau dû au sous échantillonnage

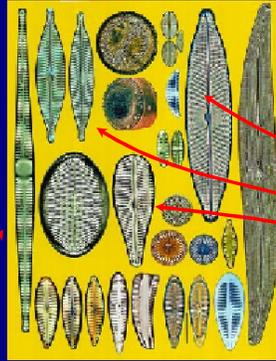


Autre exemple d'effet stroboscope

Saisie au scanner
d'une image
tramée



Autre exemple d'effet stroboscopique



inclinaison
erronée de lignes

Apparition d'ondulations lentes

Le format Jpeg

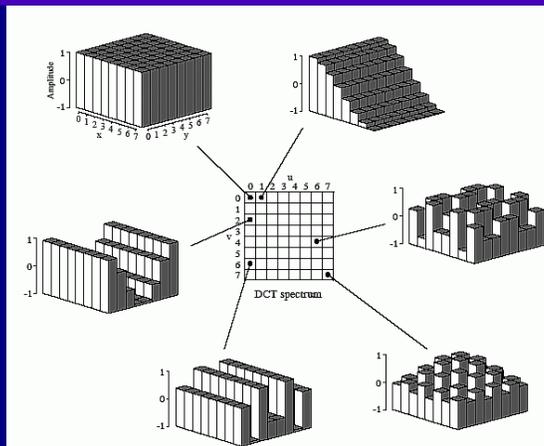
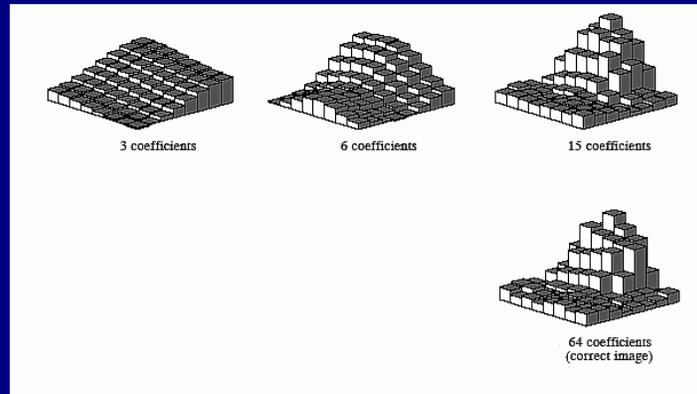
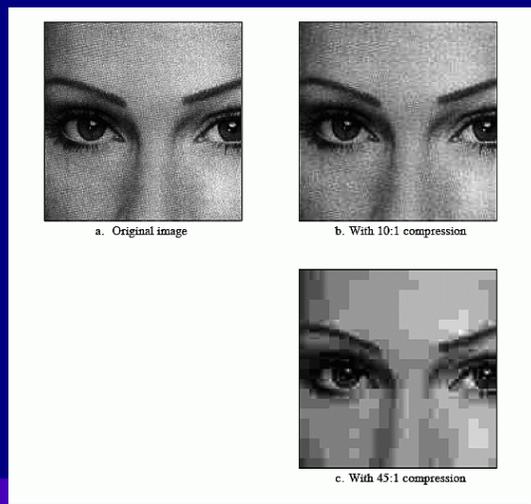


FIGURE 27-10
The DCT basis functions. The DCT spectrum consists of an 8x8 array, with each element in the array being an amplitude of one of the 64 basis functions. Six of these basis functions are shown here, referenced to where the corresponding amplitude resides.

Reconstruction d'un pavé



Effet de la compression



Introduction à la morphologie mathématique

Tristan Piolot

Plateforme d'imagerie

Institut Jacques Monod

Tel: 01442774756 ou 75784

Mail: piolot@ijm.jussieu.fr

Web: <http://ijm2.ijm.jussieu.fr/imagerie/>

Morphologie Mathématique



Traitement non linéaire apparu en France dans les années 60 (J Serra)

Fondée sur la théorie des ensembles

A, B = ensemble des pixels

$A \cup B = \{x | x \in A \text{ ou } \in B\}$ Union

$A \cap B = \{x | x \in A \text{ et } \in B\}$ Intersection

$A^c = \{x | x \notin A\}$ Complément

$A - B = \{x | x \in A, x \notin B\}$ Difference

$(A)_c = \{x | x = a + c, a \in A\}$ Translation

$\hat{A} = \{x | -x \in A\}$ Inversion

Ensemble de techniques pour traiter les images binaires. Certaines opérations sont étendues sur les niveaux de gris

Différentes opérations:

érosion, dilatation

ouverture, fermeture

squelettisation

ligne de partage des eaux...

Morphologie Mathématique

La morphologie mathématique traite une image en utilisant un ensemble structurant de centre x , de géométrie et de taille connue

L'élément structurant choisi est déplacé de façon à ce que son centre x passe par toutes les positions dans l'image à analyser

Pour chacune des positions de x , on se pose une question relative à l'union ou à l'intersection de l'élément structurant avec les objets de l'image

L'ensemble des points correspondant à une réponse positive permet de construire une nouvelle image qui constitue l'image résultat

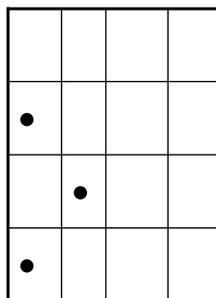
Translation

Objet noir sur fond blanc, image binaire
L'ensemble A des pixels noirs est traduit :
de 1 dans la direction horizontale, de 0 dans la direction verticale

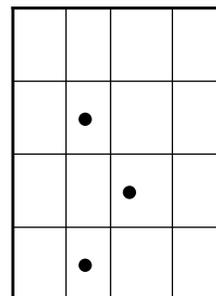
$$A = \{(0,1),(1,2),(0,3)\}$$

$$X = (1,0)$$

$$(A)x = \{(1,1),(2,2),(1,3)\}$$



A

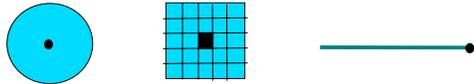


$(A)x$

Images Binaires

Élément structurant

Un élément structurant est défini par sa forme, sa taille, et son centre.



Un élément structurant peut avoir un centre dissocié

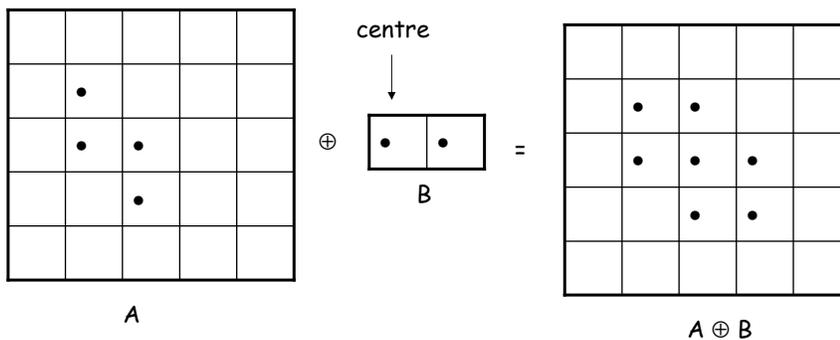
La dilatation

La dilatation est une transformation de base

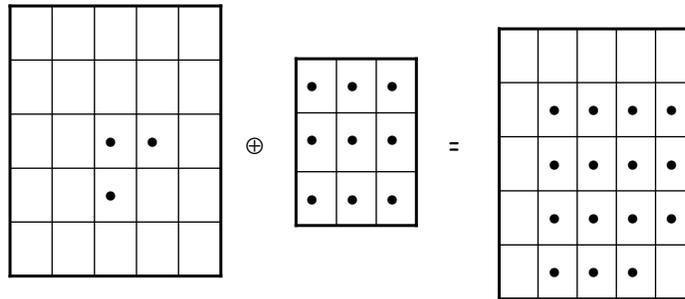
Ensemble des points pour lesquels il existe une intersection non nulle entre l'élément structurant et l'image initiale

$$A \oplus B = \{x \mid x = a + b, a \in A, b \in B\}$$

Combine deux ensembles à travers la somme de ses éléments



La dilatation



L'élément structurant utilisé est souvent un carré de 9 pixels

La dilatation:

- Ajoute un pixel sur le pourtour des formes
- Bouche les trous
- Fait se rencontrer deux formes voisines séparée par une distance moins grande que l'élément structurant

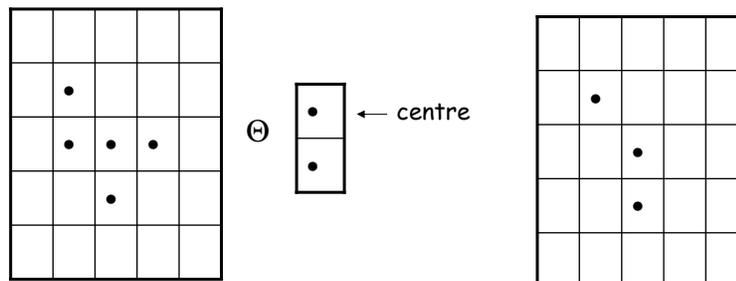
L'érosion

L'érosion est une transformation de base
C'est l'opération duale de la dilatation

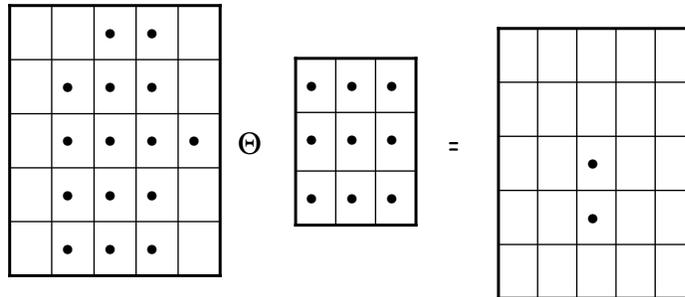
Ensemble des points pour lesquels l'élément structurant est entièrement contenu dans l'image initiale

L'érosion de A par l'élément structurant B est l'ensemble d'éléments pour qui la translation de B est contenu dans A

$$A \ominus B = \{x \mid x + b \in A, \forall b \in B\}$$

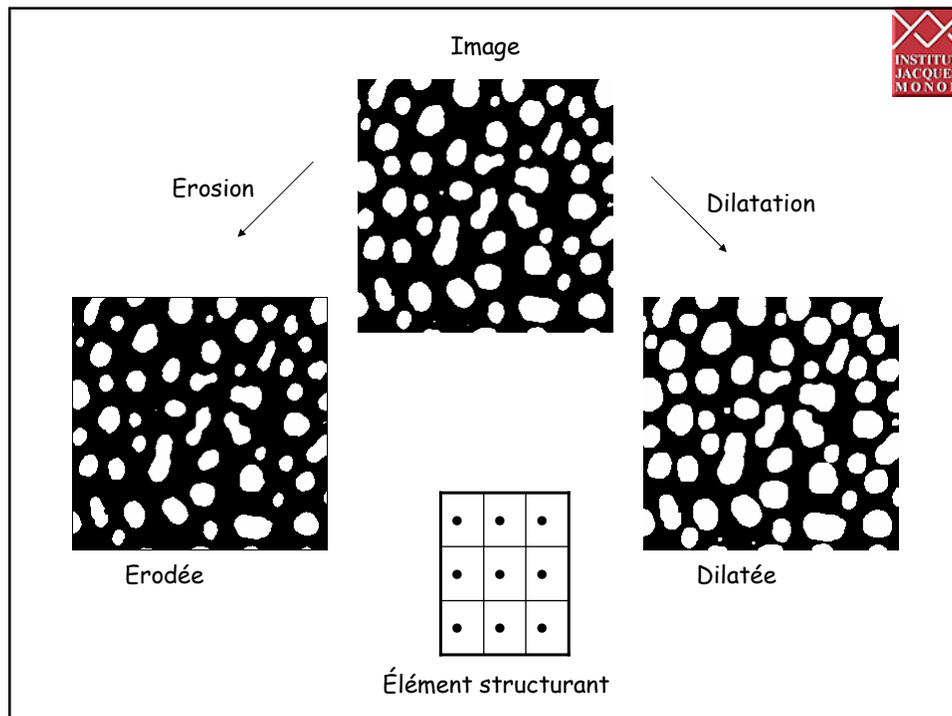


L'érosion



L'élément structurant utilisé est souvent un carré de 9 pixels

L'érosion :
Enlève des pixels sur les pourtours de la forme
Fait disparaître les objets de taille inférieure à celle de l'élément structurant
Agrandit les trous
Les objets reliés par un contact fin sont séparés



Application : détection de contour

Image des contours = image originale - image érodée

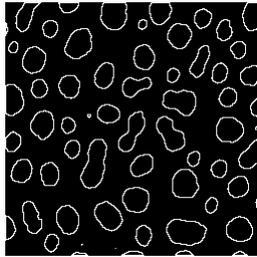


Image des contours

=

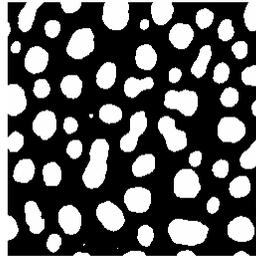


Image originale

-

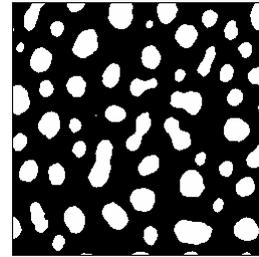
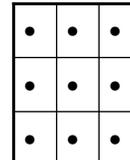


Image érodée

Élément structurant



Ouverture

L'ouverture consiste en une érosion suivie d'une dilatation avec le même élément structurant



En général, on ne retrouve pas l'ensemble de départ car une partie de la forme éliminée par l'érosion ne peut être recrée par une dilatation

L'ouverture adoucit les contours, supprime les objets plus petit que l'élément structurant

La forme des objets massif n'est pas modifiée

L'ouverture est idempotente (invariant si itérée)

Fermeture

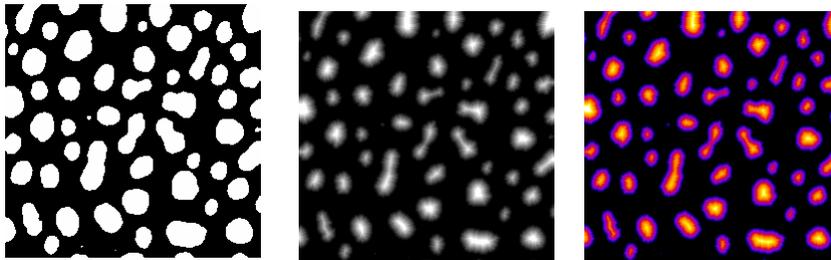
La fermeture consiste en une dilatation suivie d'une ouverture avec le même élément structurant



En général, on ne retrouve pas l'ensemble de départ
 Les fentes fines et les petits trous sont bouchés
 La forme des objets massif n'est pas modifiée
 L'ouverture est idempotente (invariant si itérée)

Fonction distance

A chaque point de l'objet est attribué une valeur de niveau de gris égale à sa distance à la frontière



Fonction distance



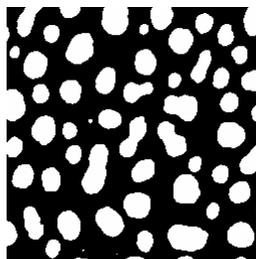
Image initiale	Image érodée	Image érodée																																																																																																																																																																																																
<table border="1"> <tr><td></td><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td></td><td></td></tr> </table>				1	1	1	1			1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			1	1	1	1			<table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td>1</td><td>1</td><td></td><td></td></tr> <tr><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td></td><td></td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td></td><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>													1	1					1	1	1	1				1	1	1	1	1	1			1	1	1	1	1	1			1	1	1	1	1	1					1	1	1	1										<table border="1"> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td>1</td><td>1</td><td></td></tr> <tr><td></td><td></td><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td>1</td><td>1</td></tr> <tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr> </table>																																						1	1						1	1	1	1							1	1								
			1	1	1	1																																																																																																																																																																																												
	1	1	1	1	1	1																																																																																																																																																																																												
1	1	1	1	1	1	1	1																																																																																																																																																																																											
1	1	1	1	1	1	1	1																																																																																																																																																																																											
1	1	1	1	1	1	1	1																																																																																																																																																																																											
1	1	1	1	1	1	1	1																																																																																																																																																																																											
1	1	1	1	1	1	1	1																																																																																																																																																																																											
		1	1	1	1																																																																																																																																																																																													
				1	1																																																																																																																																																																																													
		1	1	1	1																																																																																																																																																																																													
	1	1	1	1	1	1																																																																																																																																																																																												
	1	1	1	1	1	1																																																																																																																																																																																												
	1	1	1	1	1	1																																																																																																																																																																																												
			1	1	1	1																																																																																																																																																																																												
					1	1																																																																																																																																																																																												
				1	1	1	1																																																																																																																																																																																											
						1	1																																																																																																																																																																																											
<table border="1"> <tr><td></td><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr> <tr><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td></td><td></td></tr> </table>				1	1	1	1			1	1	1	1	1	1		1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1			1	1	1	1			<table border="1"> <tr><td></td><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td>2</td><td>2</td><td>1</td><td></td></tr> <tr><td>1</td><td>1</td><td>2</td><td>2</td><td>2</td><td>2</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>1</td></tr> <tr><td>1</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>1</td></tr> <tr><td>1</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>2</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>2</td><td>2</td><td>2</td><td>2</td><td>1</td></tr> <tr><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td></td><td></td></tr> </table>				1	1	1	1			1	1	1	2	2	1		1	1	2	2	2	2	1	1	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	1	1	2	2	2	2	2	2	1	1	1	1	2	2	2	2	1			1	1	1	1			<table border="1"> <tr><td></td><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td></td></tr> <tr><td></td><td>1</td><td>1</td><td>1</td><td>2</td><td>2</td><td>1</td><td></td></tr> <tr><td>1</td><td>1</td><td>2</td><td>2</td><td>2</td><td>2</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>2</td><td>2</td><td>3</td><td>3</td><td>2</td><td>2</td><td>1</td></tr> <tr><td>1</td><td>2</td><td>3</td><td>3</td><td>3</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>1</td><td>2</td><td>2</td><td>2</td><td>3</td><td>3</td><td>2</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td><td>2</td><td>2</td><td>2</td><td>2</td><td>1</td></tr> <tr><td></td><td></td><td>1</td><td>1</td><td>1</td><td>1</td><td></td><td></td></tr> </table>				1	1	1	1			1	1	1	2	2	1		1	1	2	2	2	2	1	1	1	2	2	3	3	2	2	1	1	2	3	3	3	3	2	1	1	2	2	2	3	3	2	1	1	1	1	2	2	2	2	1			1	1	1	1		
			1	1	1	1																																																																																																																																																																																												
	1	1	1	1	1	1																																																																																																																																																																																												
1	1	1	1	1	1	1	1																																																																																																																																																																																											
1	1	1	1	1	1	1	1																																																																																																																																																																																											
1	1	1	1	1	1	1	1																																																																																																																																																																																											
1	1	1	1	1	1	1	1																																																																																																																																																																																											
1	1	1	1	1	1	1	1																																																																																																																																																																																											
		1	1	1	1																																																																																																																																																																																													
			1	1	1	1																																																																																																																																																																																												
	1	1	1	2	2	1																																																																																																																																																																																												
1	1	2	2	2	2	1	1																																																																																																																																																																																											
1	2	2	2	2	2	2	1																																																																																																																																																																																											
1	2	2	2	2	2	2	1																																																																																																																																																																																											
1	2	2	2	2	2	2	1																																																																																																																																																																																											
1	1	1	2	2	2	2	1																																																																																																																																																																																											
		1	1	1	1																																																																																																																																																																																													
			1	1	1	1																																																																																																																																																																																												
	1	1	1	2	2	1																																																																																																																																																																																												
1	1	2	2	2	2	1	1																																																																																																																																																																																											
1	2	2	3	3	2	2	1																																																																																																																																																																																											
1	2	3	3	3	3	2	1																																																																																																																																																																																											
1	2	2	2	3	3	2	1																																																																																																																																																																																											
1	1	1	2	2	2	2	1																																																																																																																																																																																											
		1	1	1	1																																																																																																																																																																																													
Image initiale	+ érodée	+ érodée = image de distance																																																																																																																																																																																																

Érodée ultime



L'érodée ultime est le maximum de la fonction distance

			1	1	1	1									
	1	1	1	2	2	1									
1	1	2	2	2	2	1	1								
1	2	2	3	3	2	2	1				1	1			
1	2	3	3	3	3	2	1			1	1	1	1		
1	2	2	2	3	3	2	1					1	1		
1	1	1	2	2	2	2	1								
		1	1	1	1										



D'après :

Cours de Traitement d'image - Max Mignotte
Département d'informatique et de recherche Opérationnelle
(<http://www.iro.umontreal.ca/~mignotte/ift6150>)

Cours de Morphologie mathématique de M Benchaib, stage d'analyse
d'image du centre commun de quantimétrie - UCB - Lyon 1
(<http://quantimétrie.univ-lyon1.fr>)

Cours maîtrise de la segmentation d'images en microscopie d'Alain
Dieterlen, organisé par Yves Usson à l'IN3S (<http://www-timc.imag.fr/>)
et le réseau de Microscopie photonique de Fluorescence
Multidimensionnelle (<http://rtmf.m.ibl.fr/>)

Fabrice CORDELIÈRES

**Un tour guidé dans la jungle
des études de co-localisation**

Un tour guidé dans la jungle des études de co-localisation

Débroussaillage de la piste assuré par
Susanne Bolte (IFR 87 "La Plante et son Environnement"-Gif) &
Fabrice P. Cordelières (IC-Orsay)

Etude de co-localisation

Une vue d'ensemble

Qu'appelle-t-on co-localisation ?

« La co-localisation est obtenue quand deux protéines d'intérêt sont détectées sur un même pixel, telles que mises en évidence en utilisant la fréquence d'échantillonnage spatiale la plus appropriée et le même grossissement. »

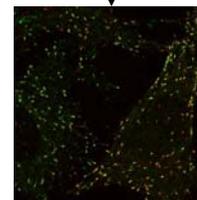
Quelles conclusions tirer de l'évaluation de la co-localisation ?

- En biologie cellulaire :
les deux protéines sont au même endroit
- Au niveau de la statistique:
à la résolution utilisée, on ne peut exclure que les deux protéines soient au même endroit.

Trois méthodologies d'évaluation de la co-localisation

Visualisation

Appliquer une LUT appropriée à chacune des images avant de les superposer



Quantification

Image en tant que collection de pixels

Recherche des pixels dont les intensités sont liées de manière linéaire entre les deux canaux

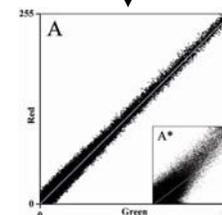
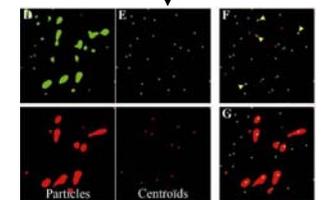


Image en tant que collection d'objets

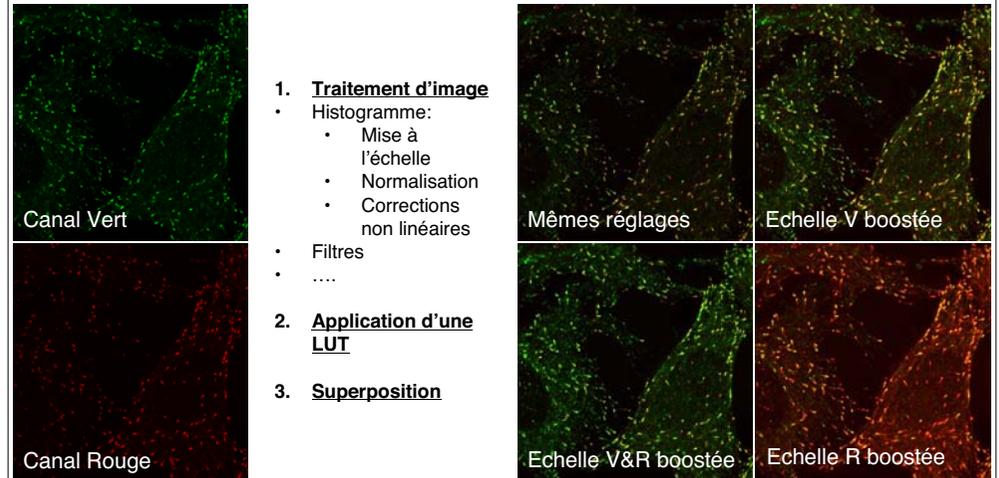
Après avoir identifié les structures, chercher une superposition partielle ou totale des objets



Visualiser la co-localisation

Visualiser la co-localisation

Quand Vert+rouge ne donne plus de Jaune....



- Traitement d'image**
 - Histogramme:
 - Mise à l'échelle
 - Normalisation
 - Corrections non linéaires
 - Filtres
 -
- Application d'une LUT**
- Superposition**

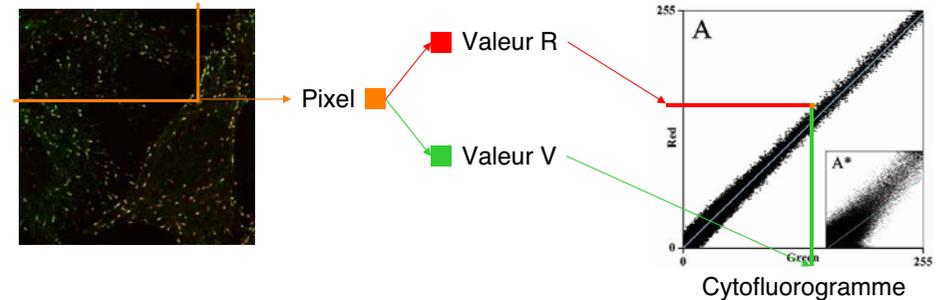
On ne peut rien conclure de l'image: elle donne une première vue générale des sites potentiels de co-localisation

Quantifier la co-localisation

Analyse reposant sur une corrélation d'intensités
(Intensity Correlation Based analysis: ICB)

Quantifier la co-localisation

ICB: cytofluorogramme & coefficient de Pearson



La linéarité de la relation est évaluée au moyen du **coefficient de Pearson**:

$$r_p = \frac{\sum_i (A_i - a) \times (B_i - b)}{\sqrt{\left[\sum_i (A_i - a)^2 \times \sum_i (B_i - b)^2 \right]}}$$

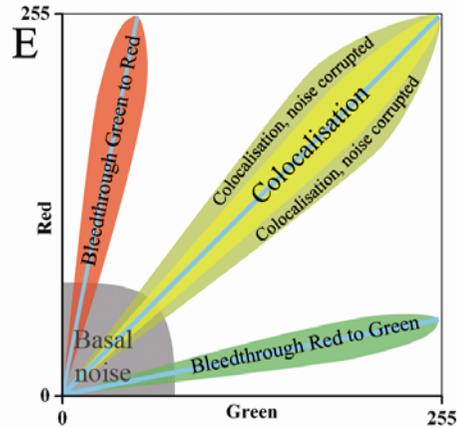
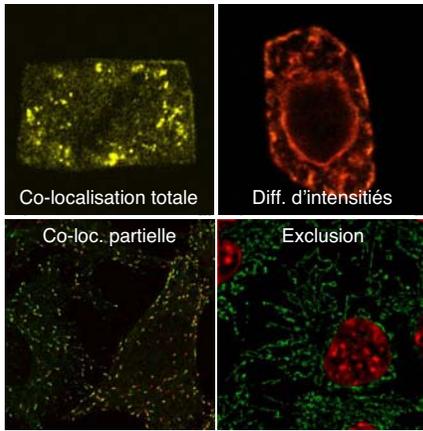
- r_p proche de -1: pas de conclusion ou exclusion
- r_p proche de 0: pas de corrélation
- r_p proche de 1: forte corrélation
- Valeurs moyennes: ???!!!

→ Impossible de conclure sur les exclusions

Manders *et al.*, (1992). J Cell Sci 103, 857-862.

Quantifier la co-localisation

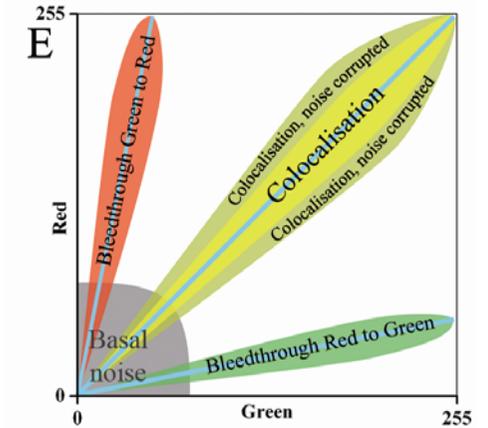
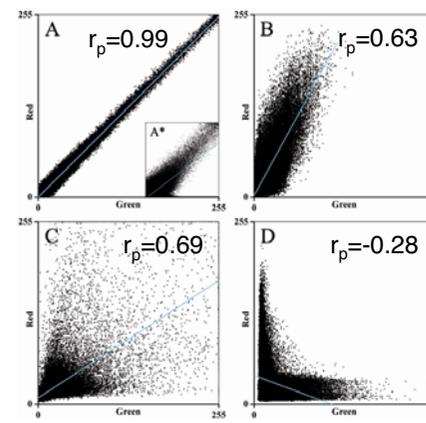
ICB: cytofluorogramme & coefficient de Pearson



Manders *et al.*, (1992). J Cell Sci 103, 857-862.

Quantifier la co-localisation

ICB: cytofluorogramme & coefficient de Pearson



Manders *et al.*, (1992). J Cell Sci 103, 857-862.

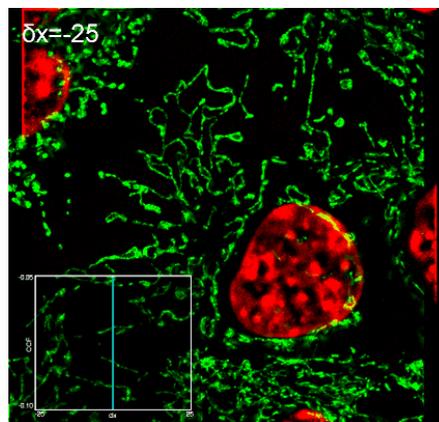
Quantifier la co-localisation

ICB: l'approche de Van Steensel

Van Steensel introduit la fonction de cross-correlation (CCF):

- Décaler l'image d'un canal par rapport à l'autre en utilisant un incrément de déplacement pré-défini
- Calculer le coefficient de Pearson
- Tracer r_p en fonction du décalage de pixels

→ Courbe en cloche = co-localisation
→ Creux = exclusion



Van Steensel *et al.*, (1996). J Cell Sci 109, 787-792.

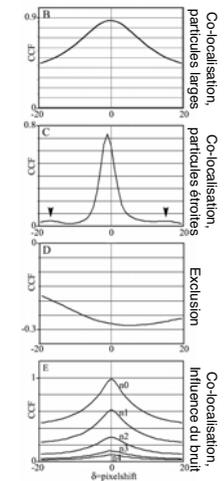
Quantifier la co-localisation

ICB: l'approche de Van Steensel

Van Steensel introduit la fonction de cross-correlation (CCF):

- Décaler l'image d'un canal par rapport à l'autre en utilisant un incrément de déplacement pré-défini
- Calculer le coefficient de Pearson
- Tracer r_p en fonction du décalage de pixels

→ Courbe en cloche = co-localisation
→ Creux = exclusion



Van Steensel *et al.*, (1996). J Cell Sci 109, 787-792.

Quantifier la co-localisation

ICB

•Le bruit est un problème majeur qui peut fortement diminuer la valeur du coefficient de Pearson

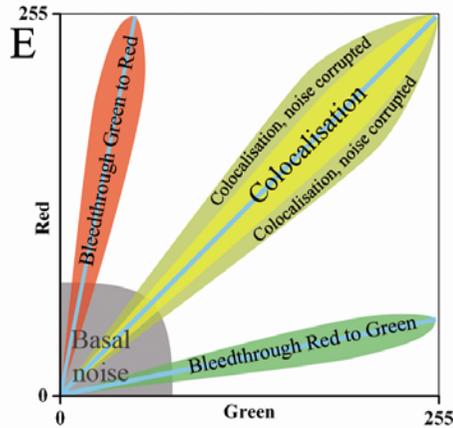
→ Sueller l'image, oui, mais comment ?!!

•Méthode de Costes: diminuer progressivement la valeur de seuil jusqu'à obtenir un r_p , calculé sur les valeurs en dessous du seuil, égal ou inférieur à zéro.

•Coefficients de Manders modifiés sont définis par :

$$M_1 = \frac{\sum_i A_{i,coloc}}{\sum_i A_i} \quad M_2 = \frac{\sum_i B_{i,coloc}}{\sum_i B_i}$$

Avec A_i , coloc égal à A_i si $B_i > thr_B$ et 0 si $B_i = 0$; et B_i , coloc égal à B_i si $A_i > thr_A$ et 0 si $A_i = 0$.



Manders et al., (1992). J Cell Sci 103, 857-862. & Costes et al. (2004). Biophys J. 86, 3993-4003.

Quantifier la co-localisation

ICB

•Le bruit est un problème majeur qui peut fortement diminuer la valeur du coefficient de Pearson

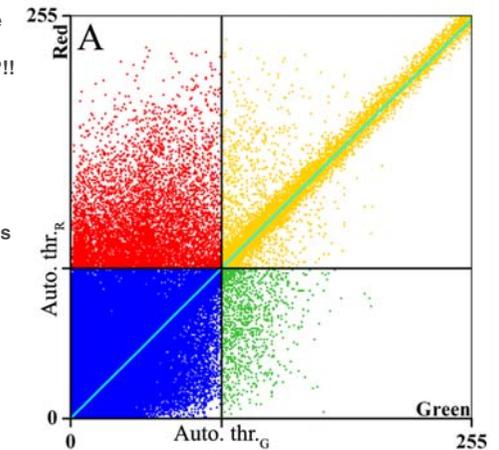
→ Sueller l'image, oui, mais comment ?!!

•Méthode de Costes: diminuer progressivement la valeur de seuil jusqu'à obtenir un r_p , calculé sur les valeurs en dessous du seuil, égal ou inférieur à zéro.

•Coefficients de Manders modifiés sont définis par :

$$M_1 = \frac{\sum_i A_{i,coloc}}{\sum_i A_i} \quad M_2 = \frac{\sum_i B_{i,coloc}}{\sum_i B_i}$$

Avec A_i , coloc égal à A_i si $B_i > thr_B$ et 0 si $B_i = 0$; et B_i , coloc égal à B_i si $A_i > thr_A$ et 0 si $A_i = 0$.

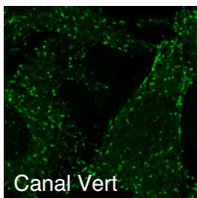


Manders et al., (1992). J Cell Sci 103, 857-862. & Costes et al. (2004). Biophys J. 86, 3993-4003.

Quantifier la co-localisation

ICB: vers une approche statistique

Alors que le coefficient de Pearson met un chiffre sur deux images, le degré de fiabilité de ce chiffre demeure obscur...

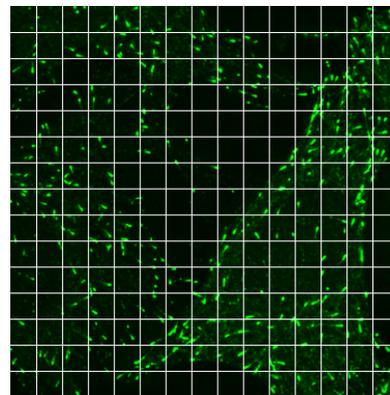


Canal Vert



Canal Rouge

1. Découper l'image verte en éléments unitaires
2. Randomiser l'image verte
3. Calculer le r_p pour chaque couple (image verte randomisée/image rouge)
4. Comparer le r_p obtenu avant randomisation à la distribution des r_p obtenus



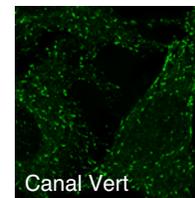
Costes propose de confronter le r_p observé à une distribution de r_p obtenue à partir d'une série de versions randomisées de l'une des images confrontées à l'autre image

Costes et al. (2004). Biophys J. 86, 3993-4003.

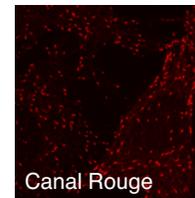
Quantifier la co-localisation

ICB: vers une approche statistique

Alors que le coefficient de Pearson met un chiffre sur deux images, le degré de fiabilité de ce chiffre demeure obscur...

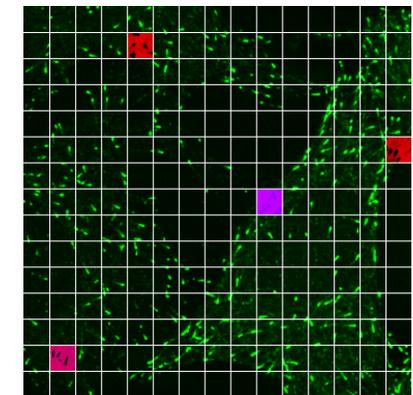


Canal Vert



Canal Rouge

1. Découper l'image verte en éléments unitaires
2. Randomiser l'image verte
3. Calculer le r_p pour chaque couple (image verte randomisée/image rouge)
4. Comparer le r_p obtenu avant randomisation à la distribution des r_p obtenus



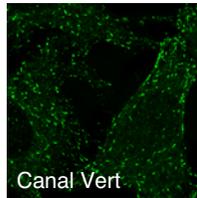
Costes propose de confronter le r_p observé à une distribution de r_p obtenue à partir d'une série de versions randomisées de l'une des images confrontées à l'autre image

Costes et al. (2004). Biophys J. 86, 3993-4003.

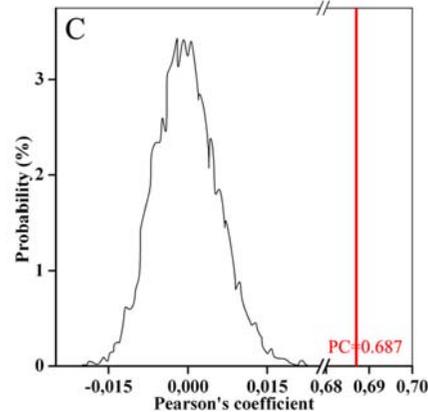
Quantifier la co-localisation

ICB: vers une approche statistique

Alors que le coefficient de Pearson met un chiffre sur deux images, le degré de fiabilité de ce chiffre demeure obscur...



1. Découper l'image verte en éléments unitaires
2. Randomiser l'image verte
3. Calculer le r_p pour chaque couple (image verte randomisée/image rouge)
4. Comparer le r_p obtenu avant randomisation à la distribution des r_p obtenus

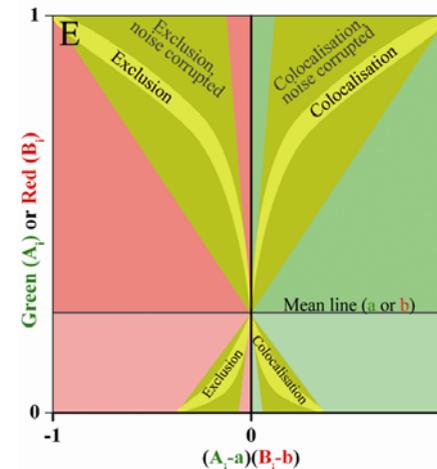


Costes propose de confronter le r_p observé à une distribution de r_p obtenue à partir d'une série de versions randomisées de l'une des images confrontées à l'autre image

Costes *et al.* (2004). Biophys J. 86, 3993-4003.

Quantifier la co-localisation

ICB: Une alternative à Pearson & Manders



La méthode de Li :

- Normaliser les intensités des deux canaux à des valeurs comprises entre 0 et 1.
- Postulat: s'il y a co-localisation, les intensités des deux canaux sont du même «côté» de la valeur moyenne d'intensité du canal considéré $(A_i - a)(B_i - b) > 0$
- Un nouvel indicateur, le ICQ: Intensity Correlation Quotient qui rend compte de la proportion de pixels dont les intensités varient de concert

→ Très bonne visualisation de l'**exclusion** et de l'**absence** de colocalisation

Li *et al.* (2004). J. Neurosci. 24, 4070-4081.

Quantifier la co-localisation

Conclusions 1/2

Coefficient de Pearson:

- Très sensible au bruit
- Peu approprié si les protéines présentent plusieurs stœchiométries d'association au sein de la même image
- Ne permet pas de mettre en évidence une exclusion
- Résultats non comparable d'un couple d'images à un autre

Coefficients de Manders:

- Seuillage difficile à réaliser, peu fiable lorsqu'il est réalisé automatiquement.
- Résultats non comparable d'un couple d'images à un autre

Méthodologie de Van Steensel:

- Un bon point de départ
- Vue plus qualitative que quantitative
- Susceptibilité vis-à-vis de la taille et de l' (an)isotropie des particules
- Facilité à révéler l'exclusion

Approche de Coste:

- La première méthode statistique
- Ne requiert pas d'image de référence à laquelle: la conclusion sur la co-localisation est absolue

Méthode de Li:

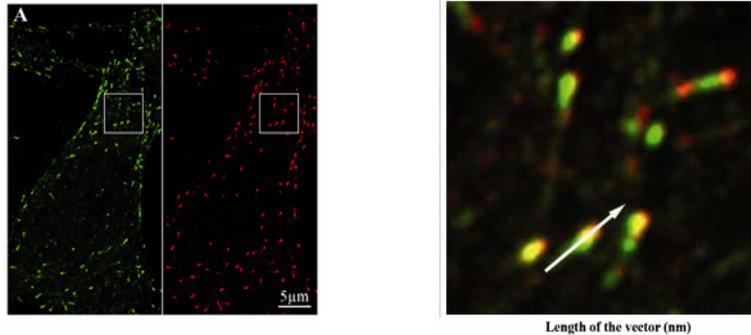
- Facilité à révéler l'exclusion/l'absence de co-localisation
- Bonne identification visuelle, directe, de la proportion de co-localisation
- Un premier pas vers la normalisation d'images

Quantifier la co-localisation

Analyse reposant sur la détection d'objets (Object Based analysis: OB)

Quantifier la co-localisation

OB: méthode des vecteurs



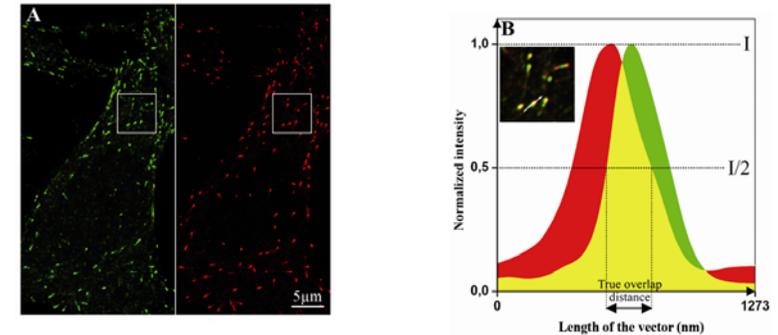
- Les structures sont pointées par l'utilisateur qui place une ligne suivant leur grand axe
- Le profil d'intensité selon ce vecteur est rapatrié pour les deux canaux
- On conclut qu'il y a co-localisation lorsque la FWHM de la zone de recouvrement des deux histogrammes est supérieure à la résolution optique de l'objectif

→ Très dépendante du positionnement du vecteur
→ Requiert l'intervention de l'utilisateur

Bolte *et al.* (2004). J Cell Sci 117, 943-954.

Quantifier la co-localisation

OB: méthode des vecteurs



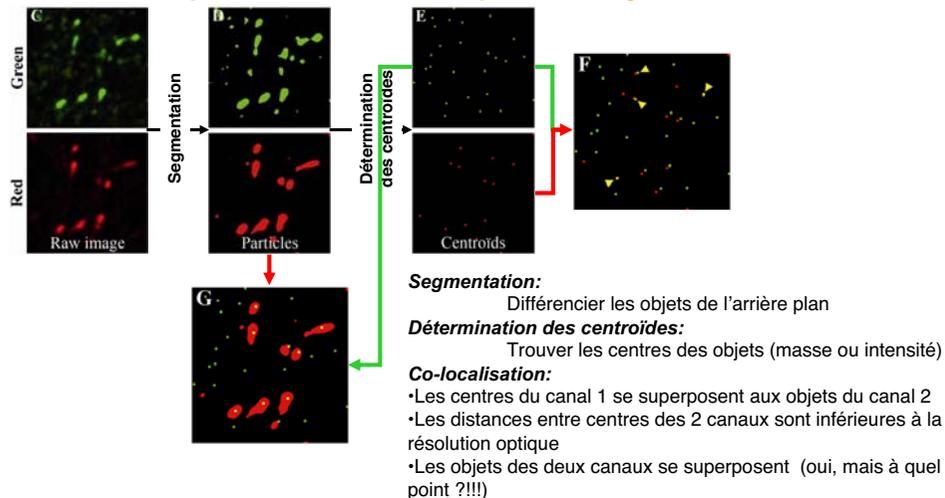
- Les structures sont pointées par l'utilisateur qui place une ligne suivant leur grand axe
- Le profil d'intensité selon ce vecteur est rapatrié pour les deux canaux
- On conclut qu'il y a co-localisation lorsque la FWHM de la zone de recouvrement des deux histogrammes est supérieure à la résolution optique de l'objectif

→ Très dépendante du positionnement du vecteur
→ Requiert l'intervention de l'utilisateur

Bolte *et al.* (2004). J Cell Sci 117, 943-954.

Quantifier la co-localisation

OB: mise à profit des caractéristiques des objets



Jaskolski *et al.* (2005). J. Neuro. Methods 146, 42-49. & Lachmanovich *et al.* (2003). J. Micros. 212, 122-131.

Quantifier la co-localisation

Conclusions 2/2

Méthode des vecteurs:

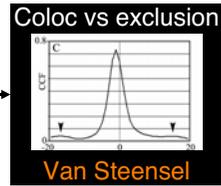
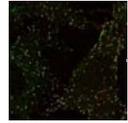
- Requiert une intervention de l'utilisateur (difficile à automatiser)
- Très dépendant de la précision de placement du vecteur
- Les structures anisotropiques peuvent se révéler difficiles à analyser
- Une des rares méthodes à prendre en compte la résolution optique du système...

Approches basées sur les caractéristiques des objets:

- Peuvent être automatisées mais...
- ...leur précision repose sur la robustesse de la méthode de segmentation
- **Objet-Objet:**
 - Difficulté d'établir un seuil de recouvrement au-delà duquel les objets sont considérés comme co-localisés
- **Objet-Centroïde & Centroïde-Centroïde:**
 - Quel centre utiliser ? Intensités or géométrique ?!
 - Peut-on utiliser ces approches si les particules sont anisotropiques ?

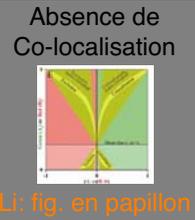
Bah, quezsétidon qu'j'va ti faire, moué ?

Inspection visuelle



Van Steensel

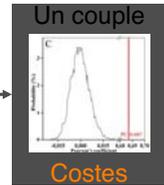
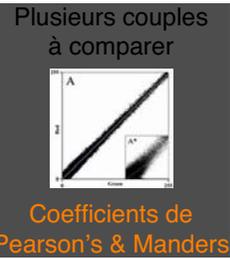
"Pas de bruit"



Nombre restreint de particules

Anisotrope
Méthode des vecteurs

Isotrope
Segmentation & quantification



Co-localisation: fin de la visite

Ressources disponibles

Logiciels:

Une multitude de greffons existent pour ImageJ:

- Colocalization: <http://rsb.info.nih.gov/ij/plugins/colocalization.html>
- Colocalization finder: <http://rsb.info.nih.gov/ij/plugins/colocalization-finder.html>
- RG2B colocalization: <http://rsb.info.nih.gov/ij/plugins/rg2bcolocalization.html>
- JACoP: <http://rsb.info.nih.gov/ij/plugins/track/jacop.html>

Littérature:

- Le manuel de Tony Collins: http://www.uhnresearch.ca/facilities/wcif/software/Plugins/colocalisation_collection.pdf
- Coming soon: Bolte S. & Cordelières F.P. (2006), J. of Microsc. (nov. ou dec. 2006)

Déconvolution : Une introduction

Christophe CHAMOT

15 Mars 2007

Table des matières

1	Quelques considérations sur les images en microscopie	2
2	Algorithmes	4
2.1	Défloutage	4
2.1.1	Renforcement et Unsharp Masking	4
2.1.2	Proches Voisins	5
2.2	Restauration	6
2.2.1	Modèle de diffraction	6
2.2.2	Filtrage inverse	6
2.2.3	Algorithmes itératifs contraints	7
2.2.4	Algorithmes statistiques	8
2.2.5	Déconvolution aveugle	9
3	Artéfacts et aberrations	9

Introduction

La déconvolution est une technique de traitement d'images utilisée pour améliorer le contraste et la résolution des images. L'idée générale est de retirer le flou induit par l'ouverture limitée des objectifs. Bien qu'historiquement, on ait commencé le défloutage sur des images en 2D — issues de l'astronomie en particulier —, on entend souvent la déconvolution en microscopie sur des images 3D. Pour obtenir une image 3D d'un échantillon, nous allons faire de vraies coupes optiques à l'intérieur de notre échantillon. En pratique, il suffit de faire le focus sur un plan donné et de prendre une image, et de recommencer jusqu'à avoir tout l'objet ...

Chaque plan est donc « pollué » par ses voisins. Il y a deux grandes stratégies pour enlever cette contribution, une approche optique et une informatique.

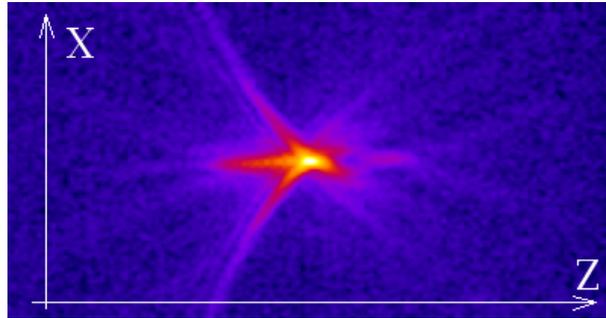
L'approche optique regroupe le confocal à balayage, où excitation et émission de lumière sont confinées par des diaphragmes (les *pinholes*); l'excitation 2-photons, où la nature non-linéaire du laser utilisé confine l'excitation au —quasi— seul plan focal ...

L'approche informatique va consister à traiter la pile d'images pour améliorer chaque tranche. C'est cette étape qui va nous intéresser ici. Elle englobe des techniques très simples de renforcement du contraste jusqu'à des modèles de restauration du contraste assez complexes.

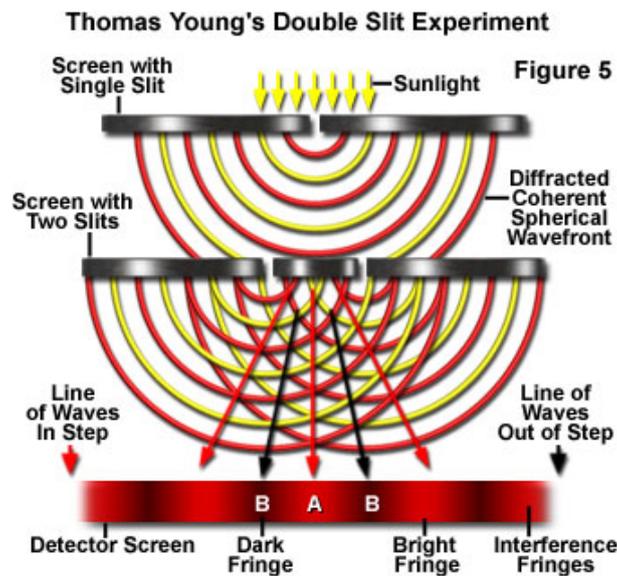
Il existe bien sûr des techniques hybrides entre ces deux approches, lorsque l'on déconvolue des images de confocal par exemple mais je pense surtout à des techniques comme l'illumination structurée où une astuce optique combinée à un calcul permet de faire des coupes nettes avec un microscope plein champ.

1 Quelques considérations sur les images en microscopie

Bien entendu, les objets ou parties d'objets qui ne sont pas au focus mais dans un plan voisin contribuent à la formation de l'image. Nous n'allons pas revoir tout le trajet optique de formation de l'image mais juste observer une image :

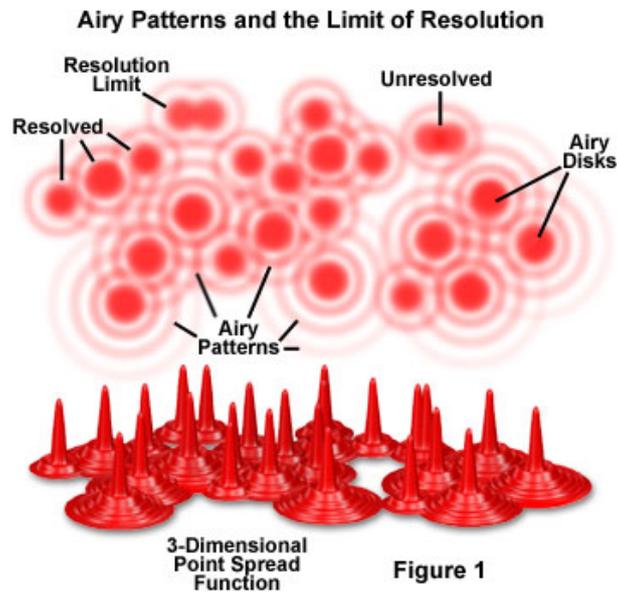


C'est une coupe selon l'axe optique (en Z, donc) de l'image d'une bille fluorescente de taille $0.17\mu m$. On voit une dispersion selon une forme de sablier (ou de diabolo, qu'on appelle PSF pour *point spread function*) qu'on peut facilement expliquer grâce à l'expérience des fentes d'YOUNG, en voici le schéma :



Les ondes lumineuses ne parcourent pas la même distance jusqu'à l'écran de visualisation. Elles interfèrent donc et, selon si elles sont en phase ou en opposition, créeront des franges claires et sombres.

L'idée est absolument la même en deux dimensions, les fronts d'ondes qui sortent d'une lentille se comportent comme autant de sources et vont donc interférer. Les lentilles des objectifs étant rondes, les figures de diffraction d'un point seront des disques. On appelle ce disque le disque d'AIKY. Il est à la base d'un des critères de résolution les plus utilisés, le critère de NYQUIST. Ce critère définit deux points comme étant résolus si la première bande sombre d'un disque touche le disque central du second disque.



2 Algorithmes

2.1 Défloutage

2.1.1 Renforcement et Unsharp Masking

On est ici dans des approches extrêmement simples, voire simplistes puisqu'on va juste chercher à réhausser les contours d'une image avec un filtre spatial, par exemple :

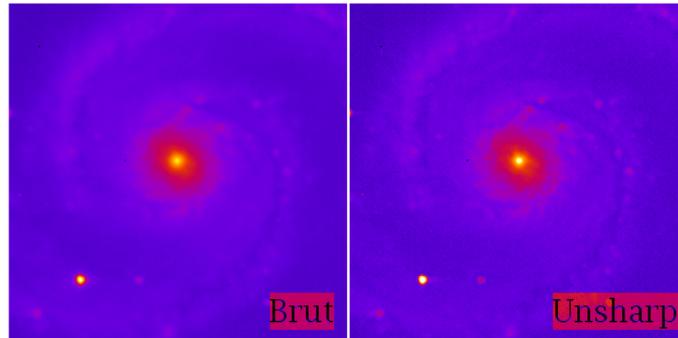
$$I_f = I_b \otimes \begin{pmatrix} -1 & -1 & -1 \\ -1 & 12 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

Les astronomes nous ont livré une méthode un peu moins simpliste, qu'on connaît sous le nom d'*unsharp masking*. Cette méthode consiste à considérer que l'image contient du flou que l'on doit retirer. Il s'agit donc d'une soustraction. Qu'est-ce qu'on va retirer ? Une pondération de l'image originale mais floutée.

$$I_f = I_b - k \cdot (I_b \otimes G(x, y))$$

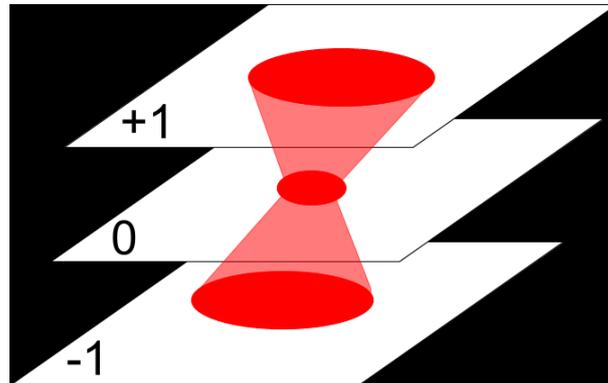
avec k comme facteur de pondération, $G(x, y)$ est en général une gaussienne.

Un exemple avec une gaussienne de rayon 4 et un $k = 0.60$:



2.1.2 Proches Voisins

Cette méthode est très fortement inspirée de la précédente, on l'applique à des acquisitions en trois dimensions. On a vu que la lumière était assez fortement dispersée le long de l'axe optique. L'idée est donc que les plans adjacents contiennent une partie floue du plan en cours.



On ne se sert dans cette méthode que des plans immédiatement adjacents. Et comme dans la méthode précédente, on fait une soustraction :

$$I_f = I_{b_0} - k \cdot \left((I_{b_{+1}} \otimes G(x, y)) + (I_{b_{-1}} \otimes G(x, y)) \right) \quad (1)$$

On retrouve cette méthode dans les logiciels sous le terme anglais de *nearest neighbors*. L'avantage est qu'elle est très rapide à mettre en œuvre et nécessite peu de mémoire. L'inconvénient majeur est que le modèle utilisé est assez simple. D'autres modèles dérivés de celui-ci ont été inventés, comme le *multi-neighbors* qui tentent de prendre en compte les contributions des autres plans :

$$\begin{aligned}
I_f = I_{b_0} - k_1 \cdot & \left((I_{b_{+1}} \otimes G_1(x, y)) + (I_{b_{-1}} \otimes G_1(x, y)) \right) \\
& - k_2 \cdot \left((I_{b_{+1}} \otimes G_2(x, y)) + (I_{b_{-1}} \otimes G_2(x, y)) \right) - \dots \quad (2)
\end{aligned}$$

On retrouve le principe de l'équation 1 en la développant sur les plans +2 et -2 et au-delà avec des pondérations (k_i) différentes et des gaussiennes différentes (σ croissants).

2.2 Restauration

L'idée directrice est d'utiliser des modèles un peu plus compliqués de dispersion de la lumière pour pouvoir faire l'opération inverse.

2.2.1 Modèle de diffraction

Pour les images de microscopie, on peut faire l'approximation que la résolution n'est limitée que par la diffraction. Le flou des images peut donc être expliqué par notre réponse instrumentale ; ici, la fameuse *psf* (voir 1 page 2). On peut donc écrire un modèle simple de formation de l'image. Si I est l'image récoltée par le microscope, O l'objet (biologique) :

$$I = O \otimes psf$$

L'expression paraît simple mais la *psf* est un objet en 3 dimensions, tout comme l'objet. Il s'agirait donc de résoudre des intégrales triples ce qui est assez peu motivant si on n'est pas matheux.

Fort heureusement, un merveilleux outil nous sauve la mise : la transformée de FOURIER. Pour une raison assez simple en fait :

$$\begin{array}{ccccc}
I & = & O & \otimes & psf \\
\text{FFT} \downarrow & & \text{FFT} \downarrow & & \text{FFT} \downarrow \\
\mathcal{F}_I & = & \mathcal{F}_O & \times & \mathcal{F}_{psf}
\end{array}$$

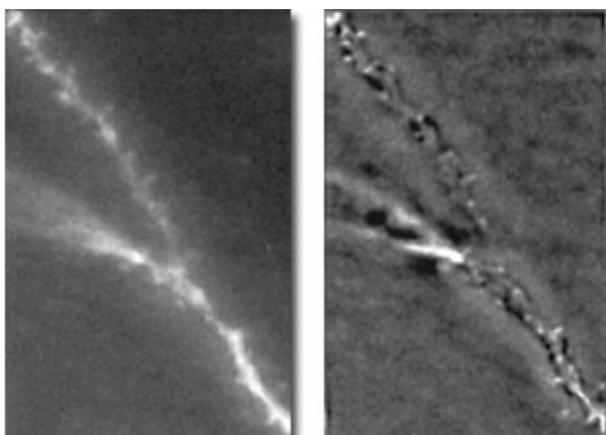
Et oui, une convolution dans l'espace normal se transforme en simple multiplication dans l'espace de FOURIER ...

2.2.2 Filtrage inverse

Évidemment, la première idée est de faire une division. En effet, compte-tenu de l'équation ci-dessus, il suffit de faire la division de la transformée de FOURIER (TF) de l'image par la TF de la *psf* pour obtenir la TF de l'objet, il suffit ensuite de retourner dans l'espace normal par une TF inverse.

Ce mode de calcul est la base d'un bon nombre de filtres, en particulier des filtres inverses régularisés. On les retrouve dans les logiciels sous le nom de filtre de WIENER, Regularized Least Squares, Linear Least Squares et TIKHONOV-MILLER Regularization.

Si ce calcul est très rapide, presque aussi rapide que les méthodes précédentes, il présente un désavantage certain. Pendant la division dans l'espace de FOURIER, les petites variations dues au bruit sont amplifiées. Le retour dans l'espace normal est donc souvent catastrophique. De plus, on peut introduire un artefact connu sous le nom de « ringing » :

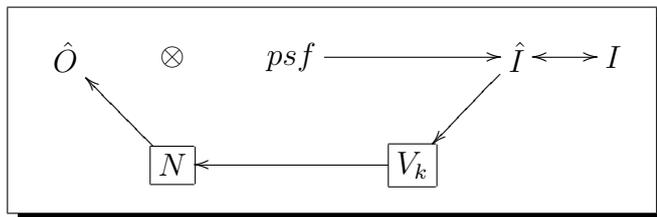


L'amplification du bruit et le « ringing » peuvent être minimisés en faisant quelques suppositions sur la nature du bruit dans l'image et la nature du signal. C'est dans cette approche qu'on retrouve le terme de *régularisation* présents dans nombre de ces filtres, on peut supposer que les objets ne comportent pas d'angles droits. Un filtre inverse régularisé peut donc être considéré comme un estimateur statistique avec des contraintes sur les objets estimés possibles. Selon les filtres, la régularisation peut se faire en une fois ou de façon itérative.

2.2.3 Algorithmes itératifs contraints

Une étape supplémentaire est franchie ici dans le but d'améliorer les performances des filtres inverses. Cette étape regroupe un certain nombre d'algorithmes tri-dimensionnels sous le terme d'*algorithmes itératifs contraints* et fonctionnent donc en cycles successifs.

Le principe de fonctionnement est assez simple :

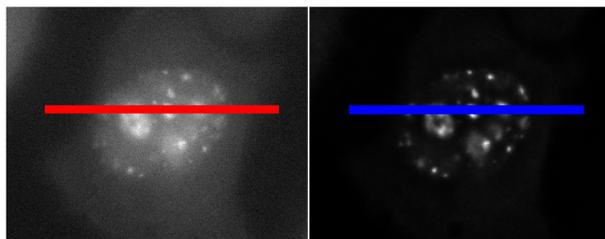


L'estimateur est convolué par la psf . Cet estimateur flou est comparé à l'image I (celle obtenue par le microscope). On calcule un vecteur de correction (V_k) qui sera appliqué à l'estimateur après une éventuelle étape de régularisation du bruit (N). Le nouvel estimateur entre dans la boucle. On cherche donc à faire converger I et \hat{I} .

Sur le plan des contraintes, on ne cherche pas seulement à minimiser les artefacts et retirer le bruit mais bien à augmenter la capacité de restauration de l'algorithme, c'est-à-dire à augmenter le rapport signal sur bruit de l'estimation de façon drastique. Une des contraintes est la non-négativité, les divisions dans l'espace de FOURIER ou les soustractions de régularisation du bruit pourrait conduire à des valeurs négatives. On gardera les anti-photons pour la physique théorique ... D'autres contraintes ont bien sûr été utilisées sur la distribution du bruit, les frontières d'objets ou autres contraintes statistiques.

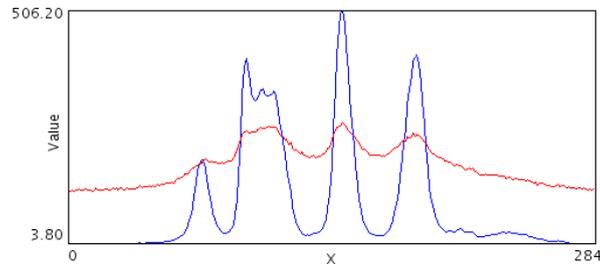
2.2.4 Algorithmes statistiques

Ces méthodes utilise la notion de vraisemblance, critère d'erreur probabilistique. On retrouve ici les algorithmes assez robustes utilisés dans la plupart des logiciels commerciaux : Maximum Likelihood Estimation (MLE), Expectation Maximum (EM, à ne pas confondre avec Maximum Entropy —ME), ...



Deux plans isolés d'une série en Z, brut et déconvolué MLE

Si on examine ce qui se passe au niveau du profil :



Les intensités de l'image déconvoluée ont été divisées par 50.

On voit bien l'effet de cet algorithme sur les profils d'intensité. Les pics sont plus étroits et plus hauts (surtout quand on sait que les intensités déconvoluées ont été divisées par 50 pour tenir sur le même graphe). En fait, on a bien augmenté le rapport signal sur bruit, le contraste donc la résolution.

Tous ces algorithmes sont assez gourmands en terme de ressources informatiques. Ils ont l'avantage d'inclure des contraintes statistiques sur le bruit (poissonnien ou gaussien) et non une simple étape de régularisation. Ils seront bien plus efficaces sur des images bruitées que les filtres inverses régularisés.

2.2.5 Déconvolution aveugle

Curieux terme que ce dernier algorithme ... En fait, il est le plus gourmand en terme de calcul mais simplifie vraiment la vie de l'utilisateur. En effet, on peut le ramener à un algorithme statistique mais un peu spécial ; on fait une première déconvolution avec une *psf* théorique. L'estimé flou de l'objet est comparé à l'image et on calcul un vecteur de correction, jusque là tout est normal. Mais on applique aussi le même traitement à la *psf*. La *psf* est ensuite estimé à chaque tour, comme l'objet.

L'avantage est de pouvoir se passer de l'acquisition assez fastidieuse des *psf* et de corriger partiellement les aberrations sphériques.

3 Artéfacts et aberrations

Les défauts qui peuvent gêner la déconvolution sont nombreux :

- problèmes liés à la *psf*
- aberration sphérique
- intervalle d'échantillonnage

- effets de bords (image et structures)
- amplification du bruit et lissage excessif
- disparitions et fausses structures
- artéfacts liés à l'échantillon

On en arrive à un aspect très important de la déconvolution qui est l'expertise. En effet, on ne peut pas se contenter d'utiliser un paramètre qui fonctionnerait pour tous les types d'images.

Pour faire ces calculs de déconvolution, qui sont —pour la plupart— des séries de convolutions en fait, nous avons fait toute une série d'hypothèses. La première est que nous serions dans un système linéaire et invariant par translation. C'est une approximation qu'on peut faire uniquement si la *psf* ne variait pas en Z ou au voisinage d'un objet, or c'est faux. D'autant plus faux que la profondeur sera grande. En pratique, il est vain d'essayer de déconvoluer des objets de plus 50μ . Il faut limiter au maximum les aberrations sphériques (choix —et prix— de l'objectif), échantillonner correctement son signal pour ne pas avoir de fréquences manquantes dans la TF et enfin *toujours* comparer l'image déconvoluée à l'image brute et remettre en doute l'estimation si des objets apparaissent ou disparaissent sans raison. Un dernier point, malgré tout le soin que l'on pourra mettre dans le choix des paramètres déconvolution, il faut rester conscient des artéfacts que l'on peut introduire par la préparation des échantillons, la fixation qui abime des bouts de microtubules et les fait apparaître punctiformes, l'aplatissement dû à la fixation au paraformaldéhyde en sont deux exemples.

Les méthodes de calcul sont puissantes mais ne doivent pas vous retirer votre libre arbitre ; au final, c'est vous qui tranchez.

Reconstruction 3D

Tristan Piolot
Plateforme d'imagerie
Institut Jacques Monod
Tel: 01442774756 ou 75784
Mail: piolot@ijm.jussieu.fr
Web: <http://ijm2.ijm.jussieu.fr/imagerie/>

Introduction à la 3D

- 1) Perception de la 3D
- 2) Représentation de l'information 3D
 - déroulement de stack
 - 2D et demi
 - Projection d'intensité maximale
 - Projection d'intensité moyenne
 - lancé de rayons
 - Volume rendering
 - Génération de surface
 - Isosurface
- 3) De la 2d à la 3D

Perception du 3D

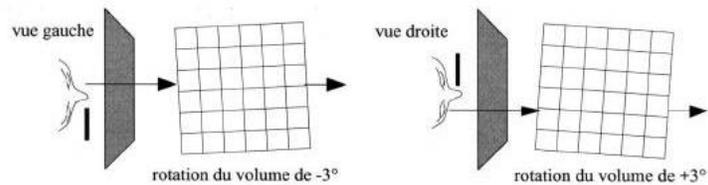
Les indicateurs physiologiques:

- accommodation (ajustement de la longueur focale du cristallin)
- convergence (angle entre l'axe de visée de chaque œil)
- Disparité binoculaire (dépend de l'angle parallaxique)
- Parallaxe du mouvement (vitesse de déplacement de l'objet)

Les indicateurs psychologiques:

- Taille de l'image rétinienne
- Perspective linéaire
- Perspective de surface
- Recouvrement
- Nuance et ombres
- Gradient de texture (lissage au loin)

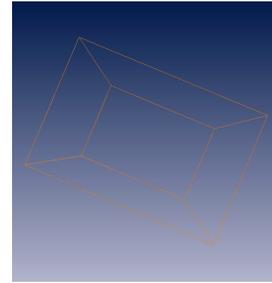
Stéréographie



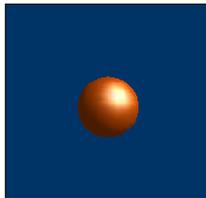
Lunettes Rouge-Cyan ou Rouge-Vert, à cristaux liquides ou à polariseurs avec projecteur d'images adapté

Parallaxe du mouvement

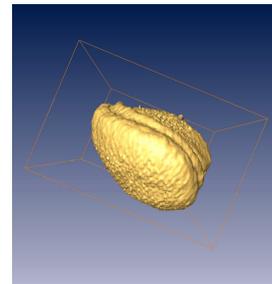
Le mouvement nous permet d'accéder à l'information 3D



Ombres



Les zones ombrées nous permettent de percevoir le volume



Qu'est-ce qu'une image 3D

Acquisition successive d'image 2D (x,y) à des positions de z différents

→ Constitution d'une pile d'image (souvent appelé stack !)

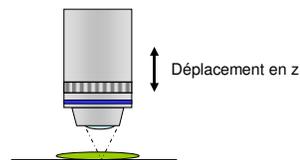
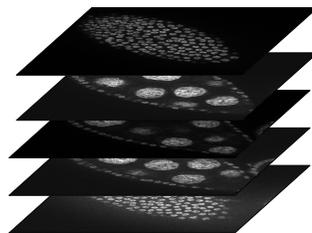
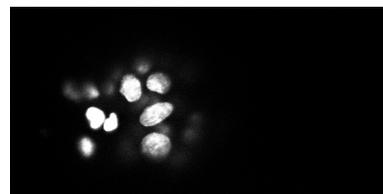


Image 3D peut être décrite par une fonction : intensité = $f(x,y,z)$

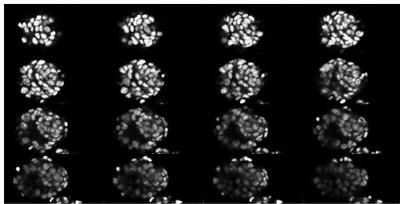
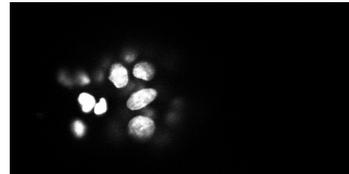
En 3D, le pixel devient voxel !!



Comment représenter une image 3D sur un support 2D ?

- Déroulement de la pile d'image

| Ne convient pas à tous les supports
| Perception du 3D difficile



- Choix de quelques plans

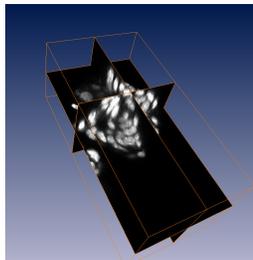
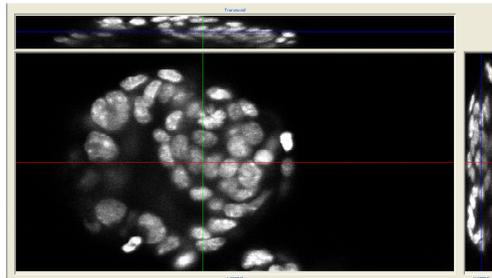
| Perte d'une partie de l'information
| Images de petite taille !!

Comment représenter une image 3D ?

- représentation en « 2D et demi »
| Ajout d'orthoSlices
| ou coupes perpendiculaires

Image en x,y
Coupe selon l'axe y - sagittale (yz)
Coupe selon l'axe x - transversale (xz)

Perte d'une partie de l'information

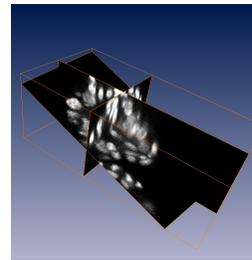


| contour d'objet et angle de vue
| particulier

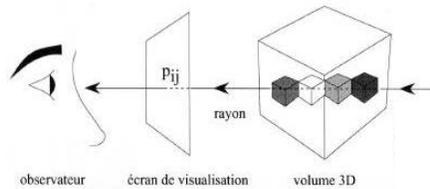
3 orthoSlices

2 orthoSlices
+ 1 obliqueSlice

Perte d'une partie de l'information



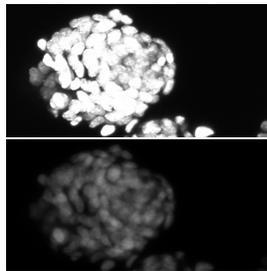
Lancé de rayons



Conversion de l'information 3D en une information 2D

Support de nombreuses transformations familières !!

- Les projections d'intensité maximale, moyenne, minimale

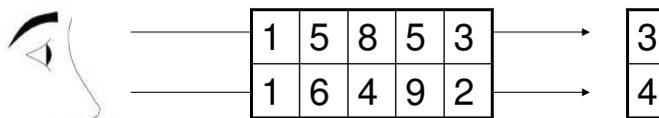


Projection d'intensité maximale:
La valeur attribué à chaque pixel correspond au maximum rencontré parmi tous les z

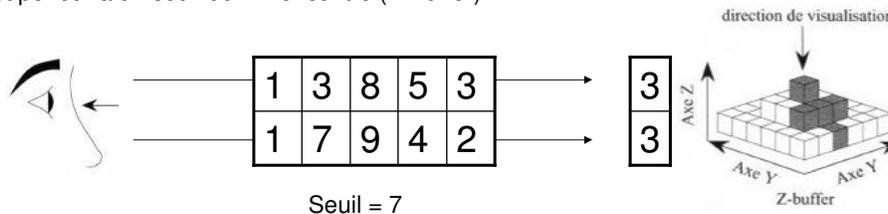
Projection d'intensité moyenne:
La valeur attribué à chaque pixel correspond à la moyenne de tous les z rencontrés

Lancé de rayons

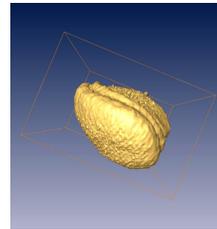
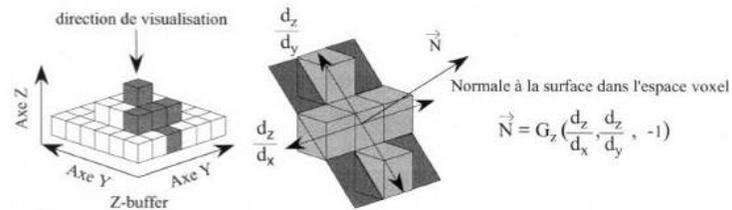
- La distance entre l'utilisateur est le pixel de la colonne d'intensité la plus élevée



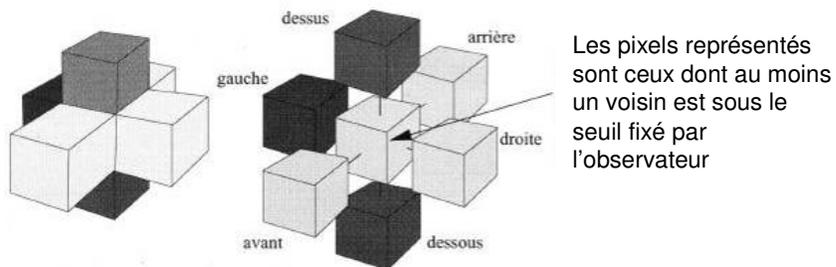
- La distance entre l'utilisateur est le premier pixel de la colonne d'intensité supérieur à un seuil défini rencontré (Z-Buffer)



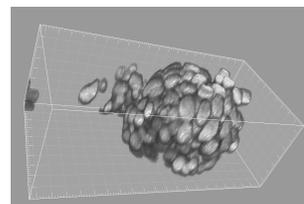
Algorithme d'ombrage



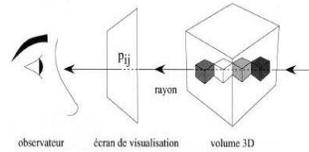
Rendu de surface



L'information située à l'intérieur des objets est perdue



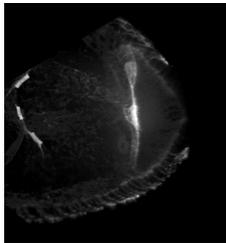
Rendu volumique



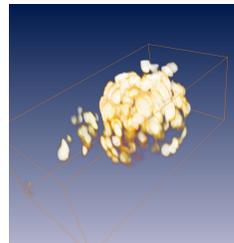
Possibilité de visualiser toutes les valeurs présentes dans le volume pour une propriété donnée

$$I_k = A_k V_k + (1 - A_k) I_{k-1}$$

$$A_k = f(V_k)$$



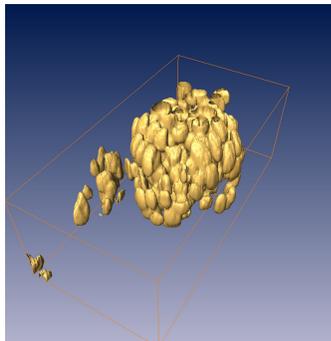
Protéine du cytosquelette



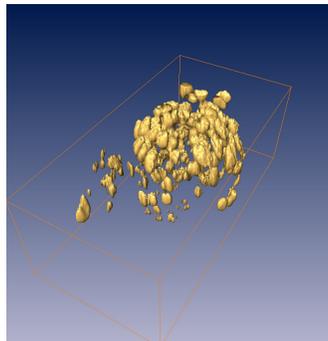
Embryon de souris

Isosurfaces

Tous les pixels dont l'intensité est supérieure au seuil choisi par l'observateur sont reliés. Dans certains cas, on peut utiliser deux valeurs de seuil : une minimale et une maximale.



Seuil niveau de gris: 160



Seuils niveau de gris:
Min: 200
Max: 230

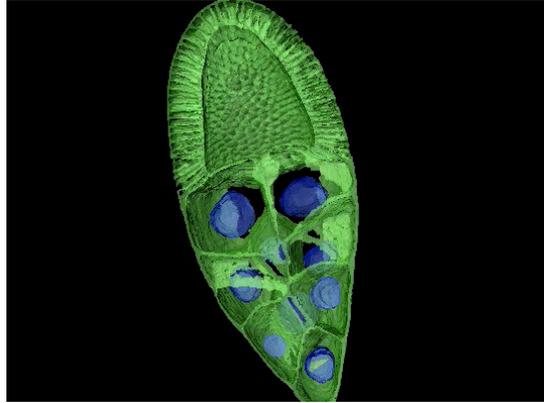


Génération de surface

Volume rendering sur
image segmentée

Chambre ovarienne de drosophile
Membrane en vert (GFP)
Noyaux en bleu (noyaux)

Segmentation par propagation
Segmentation manuelle
Volume rendering
Rotation et ombrage



Traitement et analyse d'images : de la 2D à la 3D

Difficultés liés aux systèmes d'acquisitions:

- | anisotropie du voxel
- | Atténuation du signal en profondeur

Opération morphologiques, application de filtres, opération diverses

- Passage de 4 ou 8 voisins à 6 ou 36 voisins
- Problème de volume de padding
- Comptage d'objet



Introduction aux macros sous ImageJ

Tristan Piolot

Plateforme d'imagerie

Institut Jacques Monod

Tel: 0144275784

Mail: piolot@ijm.jussieu.fr

Web: <http://ijm2.ijm.jussieu.fr/imagerie/>

Table des matières

Introduction aux macros sous ImageJ	1
Table des matières	2
Introduction	3
Premiers pas	3
Une vraie macro	4
Boucles	5
For.....	5
While	5
Do While.....	6
Variables dans les macros d'imageJ	7
Incrémentation de nom de fichier.....	8
Structures de contrôle.....	11
Utilisation de fonctions dans les macros	12
Comment faire pour déclarer une fonction.....	12
Comment faire pour appeler une fonction ?	13
Problème de numérotation et crise de la dizaine	14
Comment sauvegarder, déclarer, installer une macro ou plusieurs.....	16
La façon simple	16
Déclaration.....	16
Installation de plusieurs macros	17
Installer une ou plusieurs macro au démarrage d'ImageJ	19
Associer une touche du clavier à une macro	19
Identification des fenêtres	19
Communication avec l'utilisateur	22
print.....	23
showMessage.....	23
write	23
getString.....	23
getResults.....	23
Incorporer des commandes et des plugins dans vos macros : les limites.....	24
Transformation d'une image en tableau	24
Conclusion.....	26

Introduction

Le langage macro est relativement simple et ne nécessite pas de connaissances particulières en informatique. Dans le coin du développeur du site Image J (<http://rsb.info.nih.gov/ij/developer/>), vous trouverez toutes les ressources nécessaires à la réalisation de vos macros.

En particulier, un tutoriel, la liste des fonctions utilisables (Built-in Macro Functions) et plus de 80 exemples de macros...

(<http://rsb.info.nih.gov/ij/developer/macro/macros.html>)

Si d'aventure vous vous retrouviez malgré tout dans une impasse, il vous reste la solution simple d'utiliser la liste du forum ImageJ.. (pour vous inscrire :

<http://rsb.info.nih.gov/nih-image/list.html>)

Premiers pas

La façon la plus simple de découvrir les macros et d'utiliser le « macro-recorder ». (Plugins/Macros/Record...). Vous voyez apparaître une fenêtre ; Recorder.. dans laquelle figure à partir de maintenant toutes les actions effectuées.. Par exemple, si vous ouvrez une image, appelez FluorescentCells.tif. située dans le répertoire /home/tristan/applis/ alors apparaît la ligne suivante..:

```
open("/home/tristan/applis/FluorescentCells.tif"); si vous fermez maintenant cette image apparaît la ligne: close();
```

Il est simple de nommer cette macro en modifiant l'intitulé macro par (ex: mapremieremacro). Pour générer votre première macro, il suffit de cliquer sur Create

Avant toute autre chose, sauvez votre macro.(File/Save As/) à l'endroit qui vous paraît propice.. puis lancez la (File/Run Macro ou Ctrl R)... si tout se passe bien, vous voyez une fenêtre s'ouvrir, puis se fermer.

Quelques commentaires et remarques en vrac s'imposent.

N'oubliez pas de sauvegarder votre macro avant de la lancer, sinon, une erreur (et il y en aura !!) risquerait de « planter » ImageJ et de vous faire perdre votre précieux code..

Toutes les lignes de code se terminent par des points virgules.

Les macros sont des fichiers textes (ex: mapremieremacro.txt), il est donc possible de les ouvrir dans n'importe quel éditeur de texte.. (wordpad, simpletext..)..

Le langage macro est un langage interprété (pas de compilation comme en c ou en java) qui ressemble par sa structure et certaines de ses fonctions à du java.

Une vraie macro

Commençons par un exemple simple, la réalisation d'une projection d'intensité maximale sur un stack.. facile.. vous ouvrez ImageJ, vous ouvrez le stack (Fichier/Ouvrir), vous faites la projection d'intensité (Image/Stack/Z Project) vous sélectionnez Max Intensity et voilà... vous sauvez ça sous forme d'un fichier tif

Alors que vous pensiez être débarrassés de cette lourde tâche, vous découvrez qu'il y a en réalité non pas un stack mais 9 correspondants à autant de point de temps... Vous n'avez plus le choix, il va vous falloir réaliser une macro.

Comme précédemment, lancez le Macro Recorder, ouvrez le premier stack de la série, puis réalisez la projection d'intensité et sauvez dans le dossier approprié, (« Résultats », dans cette exemple).

Puis fermez le stack et la projection.

Le code obtenu devrait ressembler à ça:

```
open("/home/tristan/Desktop/Essai/Demo1.stk");  
run("Z Project...", "start=1 stop=25 projection='Max Intensity'");  
run("Tiff...", "save=/home/tristan/Desktop/Essai/Resultats/Demo1.tif");  
close();  
close();
```

Quelques commentaires:

On remarque que dans Z-Project sont définis tous les paramètres indispensables à l'opération (l'image initiale, l'image finale, le type de projection)

Il s'agit ici d'un fichier .stk, mais ça ne change rien à l'histoire, le résultat est stocké en tif.

Les deux `run("Close")` à la fin correspondent à la fermeture du stack et de l'image.

Si maintenant vous créez, sauvez et lancez cette macro, vous parvenez à refaire ce qui vient d'être fait à la main. Comment faire maintenant pour réaliser le passage au stack suivant..? Il va falloir créer une boucle.

Boucles

Une boucle est une structure permettant de répéter plusieurs fois la même séquence d'événements

Dans le langage macros d'ImageJ, il existe trois type de boucles: for, while et do while

For

for (initialisation; condition; incrément)

{instructions}

La boucle « for » exécute une liste d'instructions un nombre de fois spécifié.

L'instruction d'initialisation n'est lue qu'une fois au début de la boucle.

La condition est évaluée au début de chaque itération.

L'incrémentation à lieu après chaque itération

Exemple:

```
For (i=0; i<10; i++)
```

```
{
```

```
print(i);
```

```
}
```

Il n'y a pas de point virgule à la fin de cette ligne car l'instruction ne finit qu'au crochet fermé.. }

L'instruction print (i) permet l'affichage de la valeur de la variable i dans la fenêtre log

Fin de l'instruction

Lorsqu'il n'y a qu'une instruction dans la boucle, alors il n'est pas nécessaire de mettre les crochets

```
for (i=0; i<10; i++)
```

```
print(i);
```

While

```
while (condition)
```

```
{instructions}
```

Les instructions dans la boucle « while » ne sont traitées que si la condition est vraie.

Dès que la condition devient fausse la boucle « while » est arrêtée

Avec ce type de boucle, il est possible que le bloc d'instruction ne soit jamais traité

Exemple:

<code>i = 0;</code>	Initialisation de la variable i
<code>while (i<10) {</code>	Pas de point virgule car l'instruction ne finit qu'au crochet fermé.. }
<code> print(i);</code>	L'instruction print (i) permet l'affichage de la valeur de la variable i dans la fenêtre log
<code> i = i + 1;</code>	Incrémentatation de i
<code>}</code>	Fin de l'instruction

Do While

```
do {  
instructions  
} while (condition);
```

La condition n'est pas évaluée au début de la boucle mais à la fin.. le bloc d'instruction est au moins traité une fois.

Exemple:

<code>i = 0;</code>	Initialisation de la variable i
<code>do {</code>	Pas de point virgule car l'instruction ne finit qu'au crochet fermé.. }
<code> print(i);</code>	L'instruction print (i) permet l'affichage de la valeur de la variable i dans la fenêtre log
<code> i = i + 1;</code>	
<code>} while (i<=10)</code>	Fin de l'instruction + test de la condition

Il suffit maintenant d'inclure le bloc d'instructions de notre macro dans une boucle for

```
for (i=1; i<=9; i++){  
open("/home/tristan/Desktop/Essai/Demo1.stk");  
run("Z Project...", "start=1 stop=25 projection='Max Intensity'");  
run("Tiff...", "save=/home/tristan/Desktop/Essai/Resultats/Demo1.tif");  
close();  
close();  
}
```

La première ligne constitue le début de la boucle.. La variable *i* est initialisée à 1, la boucle s'arrêtera lorsque *i* aura une valeur supérieure à 9. A la fin de chaque itération la valeur de *i* est incrémentée de 1. Le bloc d'instruction contenu dans la boucle for va être traité 9 fois.. avec des valeurs de *i* allant de 1 à 9..

Si vous lancer cette macro, vous constaterez avec effroi quelle effectue effectivement 9 projections d'intensité maximale, mais 9 fois sur le même stack: Demo1.stk... il va donc falloir incrémenter le nom de fichier qui nous intéresse... avant d'aborder ce moment difficile, sources de très nombreuses erreurs voyons rapidement les variables dans le langage macro..

Variables dans les macros d'imageJ

Les variables dans les macros d'imageJ n'ont pas de type défini et non pas besoin d'être déclarées..

Ceci signifie que vous n'avez pas besoin de préciser, avant d'utiliser une variable, le type de données que vous entendez lui affecter.. (à la différence du C ou du Java..). L'interpréteur réalise leur existence au moment où il les rencontre dans le code, et cela lui suffit.. L'initialisation de la variable est faite au moment de son affectation..

Une variable peut être un nombre, une chaîne de caractères, un tableau..

Exemple:

```
v=15;
print(v);
```

v est un nombre

```
v="une chaîne de caractères";
print(v);
```

v est une chaîne de caractère

```
v = newArray(10, 20, 50);
for (i=0; i<=2; i++)
print(v[i]);
```

v est un tableau
La fonction `newArray` permet d'affecter ce tableau. Le premier élément de ce tableau est l'élément 0
`v[i]` signifie l'élément `i` du tableau `v`

Le nom des variables est sensible à la casse.. les variables `v` et `V` sont totalement différentes

Il est souvent préférable de donner aux variables des noms plus explicites que `v...`

Incrémentation de nom de fichier

Il suffit d'utiliser substituer une partie du nom de fichier par la valeur de la variable...

Ainsi, la ligne :

```
open("/home/tristan/Desktop/Essai/Demo1.stk");
```

est transformée en :

```
open("/home/tristan/Desktop/Essai/Demo"+i+".stk")
```

avec `i` allant de 1 à 9, il en est de même pour la ligne

```
run("Tiff...", "save=/home/tristan/Desktop/Essai/Resultats/Demo1.tif");
```

```
run("Tiff...", "save=/home/tristan/Desktop/Essai/Resultats/Demo"+i+".tif");
```

La totalité du code devient donc:

```
for (i=1; i<=9; i++){
```

```
open("/home/tristan/Desktop/Essai/Demo"+i+".stk")
run("Z Project...", "start=1 stop=25 projection='Max Intensity'");
run("Tiff...", "save=/home/tristan/Desktop/Essai/Resultats/Demo"+i+".tif");
close();
close();
}
```

Vous pouvez lancer cette macro à condition d'avoir des fichiers .stk nommés Demo1.stk à Demo9.stk

Remarques en vrac:

Les noms de fonctions, comme les variables, sont sensibles à la casse.. for existe.. For n'existe pas..

Cette macro ne fonctionne que pour des fichiers Demo1.stk à Demo9.stk situés dans le répertoire `/home/tristan/Desktop/Essai/`, mais rien ne nous empêche de modifier à la main, le chemin et le nom des fichiers dans un l'éditeur de textes.

Les opérateurs du langage des macros d'imageJ

opérateurs	priorité	Description
++	1	incrément pré ou post
--	1	décrément pré ou post
-	1	moins
!	1	complément booléen = NON logique
*	2	multiplication
/	2	division
%	2	modulo = reste
&	2	et (entre bits)
	2	ou (entre bits)
^	2	xor = OU exclusif

opérateurs	priorité	Description
<<, >>	2	déplacement gauche et droite
+	3	addition ou concaténation de chaînes
-	3	soustraction
<, <=	4	inférieur, inférieur ou égal
>, >=	4	supérieur, supérieur ou égal
==, !=	4	égal, différent
&&	5	ET booléen = ET dans if()=Et logique
	5	OU booléen = OU dans if()= Ou logique
=	6	affectation
+=, -=, *=, /=	6	affectation avec opération

Vous n'utiliserez probablement pas tous ces opérateurs, voilà une liste et quelques commentaires sur ceux que l'on rencontre le plus souvent..

i++ signifie $i = i + 1$, i-- signifie $i = i - 1$

- : soustraction, * : multiplication, / : division

! : si a est vrai alors !a est faux...

% : le reste de la division: $5 \% 2 = 1$

^ : OU exclusif : $A \wedge B$: Vrai si A vrai ou bien B vrai, mais pas les deux...

+ addition ou concaténation des chaînes: "vive"+"ImageJ" -> "viveImageJ"

<, <=, >, >= inférieur, inférieur ou égale, supérieur, supérieur ou égal

==, != : égal et différent: $A = B$: vrai si A est égal à B, $A != B$: vrai si A est différent de B

&&: $A \& B$ vrai si A et B sont vrais, ||: $A || B$ vrai si A ou B sont vrai, ou les deux

=: affectation : $A = 5$: la valeur de A est 5

+=, -=, *=, /= : opération avec affectation: $A += 5$ signifie $A = A + 5$

Structures de contrôle

Instructions If/Else

L'instruction « if » permet le traitement conditionnel d'une partie du code. Ainsi, dans la structure :

```
if (condition) {  
instructions;  
}
```

Le bloc d'instructions entre crochets ne sera exécuté que si la condition est vraie.

Il est également possible d'associer à cette structure, un bloc d'instructions à effectuer, si la condition n'est pas vérifiée, ceci en utilisant l'instruction « else ».

```
if (condition) {  
instructions_A;  
} else {  
instructions_B;  
}
```

Il est possible de sérier les instructions « else » afin d'augmenter le nombre de cas envisagé en utilisant l'association des instructions « else » et « if »:

```
if (condition_A)  
instructions_A;  
else if (condition_B)  
instructions_B;  
else if (condition_C)  
instructions_C;  
else  
instructions_D;
```

Si aucune des conditions A, B et C n'est vraie alors ce sont les instructions D qui seront exécutées.

Comme pour les boucles, les crochets après la condition ne sont nécessaires que s'il y a plus d'une ligne d'instruction.

Utilisation de fonctions dans les macros

L'utilisation de fonctions présente de nombreux avantages. Les fonctions permettent de sortir du code principal une partie du code remplissant une tâche précise et augmente ainsi la lisibilité du programme. Par ailleurs, ces fonctions peuvent être appelées à plusieurs endroits dans le programme, évitant de répéter inutilement des parties de code.

Encore mieux, comme nous le verrons dans le paragraphe suivant, il est possible de mettre plusieurs macros dans le même fichier et d'utiliser les mêmes fonctions pour les différentes macros.

Comment faire pour déclarer une fonction

Imaginons une fonction qui détermine le nombre de pixels dans une image.

Depuis le programme principal, chaque fois que vous aurez besoin de déterminer le nombre de pixel d'une image, vous appellerez cette fonction en lui passant un argument, correspondant à la référence de l'image en question, et cette fonction déterminera et vous renverra le nombre de pixels.

La syntaxe utilisée est la suivante:

```
function MaFonction(argument){  
corps de la fonctions;  
return resultat ;  
}
```

un exemple

<code>function CalculTailleImage(MonImage){</code>	Déclaration de la fonction. Cette fonction reçoit un argument prenant pour nom MonImage à l'intérieur de la fonction. Cet argument est l'identifiant de l'image qui nous intéresse
<code>selectImage(MonImage);</code>	<code>selectImage(MonImage)</code> active l'image MonImage
<code>y=getHeight();</code>	Récupère la hauteur de l'image
<code>x=getWidth();</code>	Récupère la largeur de l'image
<code>TailleImage=x*y;</code>	Calcul le nombre de pixels et affecte le résultat à la variable TailleImage
<code>return TailleImage;</code>	Retourne le nombre de pixel
<code>}</code>	

Les fonctions `selectImage()`, `getHeight()`, `getWidth()` sont des fonctions imageJ, décrites à l'adresse suivante:

<http://rsb.info.nih.gov/ij/developer/macro/functions.html>

Remarque toutes les fonctions ne renvoient pas un résultat. Certaines ne font par exemple que modifier une image. Dans ce cas, la dernière ligne est un « return » sans variable.

Comment faire pour appeler une fonction ?

Si la fonction retourne une valeur:

`variable = x`

`valeur de retour = MaFunction(variable)`

Si la fonction ne retourne pas de valeur:

`variable = x;`

`MaFunction(variable);`

A noter que l'appel de la fonction est suivi d'un point virgule.

Dans les deux cas, la variable est passée comme argument à la fonction.

un exemple

`MonImage=GetImageID;`

Récupération de l'identifiant de l'image d'intérêt

`TailleDeMonImage=CalculTailleImage(
MonImage);`

Appel de la fonction et passage de l'identifiant de l'image d'intérêt sous forme d'argument. La valeur de retour de la fonction est affectée à la variable TailleDeMonImage

Problème de numérotation et crise de la dizaine

La série de fichiers DemoX.stx ne comportait que neuf éléments Concrètement, il est fréquent de rencontrer des séries de fichiers bien plus importantes.

La numérotation de fichier de 1 à 50 s'effectue en général de la façon suivante: Demo01, Demo02.. Demo09, Demo10 .. Demo11.. Demo19.... jusqu'à Demo50.

Le problème associée à ce ty pe de construction de nom de fichier est le suivant:

La ligne utilisée précédemment pour l'ouverture itérée des fichier était la suivante:

```
open("/home/tristan/Desktop/Essai/Demo"+i+".stk")
```

avec i allant de 1 à 9... Dans l'exemple présent, les noms de fichiers vont de 01 à 50.. ImageJ ne pourra pas trouvé Demo1 qui n'existe pas..

Si vous ajouter un 0 :

```
open("/home/tristan/Desktop/Essai/Demo0"+i+".stk")
```

ImageJ trouvera les 9 premiers éléments de la série, mais ne trouvera pas Demo10 puisqu'il le cherchera sous le nom : Demo010...

Plusieurs solutions sont possible pour lutter contre l'adversité de la situation

La première consiste à recopier plusieurs fois le code. Cette solution peut élégante est assez pratique..

```
for (i=1; i<10; i++){
open("/home/tristan/Desktop/Essai/Demo0"+i+".stk")
run("Z Project...", "start=1 stop=25 projection='Max Intensity'");
run("Tiff...", "save=/home/tristan/Desktop/Essai/Resultats/Demo0"+i+".tif");
close();
close();
}
for (i=10; i<51; i++){
open("/home/tristan/Desktop/Essai/Demo"+i+".stk")
run("Z Project...", "start=1 stop=25 projection='Max Intensity'");
run("Tiff...", "save=/home/tristan/Desktop/Essai/Resultats/Demo"+i+".tif");
close();
close();
}
```

Dans cet exemple, les fichiers 1 à 9 sont ouverts par la première boucle, les fichiers 10 à 50 sont ouverts par la deuxième boucle.. dans lequel le fameux 0 a été supprimé. Une autre façon de faire, plus élégante consiste à créer une fonction qui va gérer le changement de dizaine... et qui évidemment sera utilisable pour toutes les macros que vous développerez par la suite

```
for (i=1; i<51; i++){
a=Rename(i);
open("/home/tristan/Desktop/Essai/Demo"+a+".stk")
run("Z Project...", "start=1 stop=25 projection='Max Intensity'");
run("Tiff...", "save=/home/tristan/Desktop/Essai/Resultats/Demo0"+a+".tif");
close();
close();
}
```

```
function Rename (i) {  
if (i<10) a= 0+"i";  
if (i>=10) a="i";  
return a;  
}
```

Dans cet exemple la gestion de la crise de la dizaine est prise en charge par la fonction Rename qui renvoie la variable: a.. lorsque i va de 1 à 50, a va de 01 à 09 puis de 10 à 50.. il n'a plus de problème de 0..

Cette construction un peu compliquée, permet de gérer également sans difficultés le passage des centaines.. des milliers... et bien évidemment peut être appelée à différents endroits de cette macro ou d'une autre

Comment sauvegarder, déclarer, installer une macro ou plusieurs

Il existe plusieurs façons de sauvegarder une macro.

La façon simple

Le plus simple consiste à sauver le fichier texte dans le répertoire de votre choix.. Puis, lorsque vous désirez à nouveau utiliser cette macro, il suffit d'ouvrir le fichier dans l'éditeur de texte d'ImageJ(Plugin/Edit), de sélectionner votre macro, et de la lancer (File/Run Macro ou Ctrl R).

Déclaration

Une autre façon de faire est de « déclarer les macros ». Pour cela, il suffit de rajouter dans l'éditeur, en tête de votre code la ligne: macro "LeNomDeMaMacro" { et de fermer les crochets à la fin de la macro.. }

un exemple

```
macro "MaMacroEstCool" {
    for (i=1; i<=9; i++){
        open("/home/tristan/Desktop/Essai/Demo
        1.stk");
        run("Z Project...", "start=1 stop=25
        projection='Max Intensity'");
        run("Tiff...",
        "save=/home/tristan/Desktop/Essai/Resul
        tats/Demo1.tif");
        close();
        close();
    }
}
```

Déclaration de la macro dont le nom est "MaMacroEstCool"; ouverture de cette macro :

ouverture de la boucle for

fermeture de la boucle for

fermeture de la macro

Sauvez ensuite cette macro dans un fichier texte, (dans cet exemple:MaMacroEstCool.txt) allez dans Plugins/Macros/Install, puis choisissez le fichier texte de votre macro.

Si maintenant vous allez dans Plugins/Macros vous voyez apparaître sous Record le nom de votre macro.

Il est important de noter que:

Si vous modifiez le .txt de votre macro, il faudra la réinstaller si vous désirez que ces modifications soient prises en compte..

Si maintenant vous installez une autre macro, en suivant la même procédure, celle-ci se substituera à la macro précédente

Installation de plusieurs macros

Le seul moyen d'installer plusieurs macros est de les déclarer dans le même fichier

texte.

Ainsi si vous sauvez le fichier `MesMacros.txt` suivant:

```
macro "MaMacroPréférée1" {  
  blablabla1..;  
}
```

```
macro "MaMacroPréférée2" {  
  blablabla2...;  
}
```

Après installation, celles-ci apparaissent dans le menu Plugins/Macros Il est aussi tout à fait possible de déclarer des fonctions comme expliqué précédemment dans le même fichier et de les appeler depuis les différentes macros, comme dans l'exemple suivant:

```
macro "MaMacroPréférée1" {  
  blablabla1_1;  
  glups=MaFonction(Variable1);  
  blablabla1_2;  
}  
macro "MaMacroPréférée2" {  
  blablabla2_1;  
  grrr=MaFonction(Variable2);  
  blablabla2_2;  
}  
MaFonction(argument){  
  blablablacommunauxdeuxfonctions;  
  return Resultat;  
}
```

Cette structure est un peu compliquée à mettre en place mais très simple à utiliser..

En pratique, le mieux est d'avoir deux fichiers textes. Le premier (`développement.txt`)

dans lequel se retrouvent toutes les fonctions que vous avez déjà créées ou récupérées que vous utiliserez pour réaliser de nouvelles macros. Le second, (*MesMacros.txt*, dans lequel seront toutes vos macros terminées et fréquemment utilisées).

Il est inutile d'installer *développement.txt* le mieux est de l'éditer dans ImageJ afin de pouvoir utiliser le touche Run Macro évitant la réinstallation après chaque modification.

En revanche, il peut être utile d'installer *MesMacros.txt*, voir même, de les installer au démarrage d'ImageJ.

Installer une ou plusieurs macro au démarrage d'ImageJ

Rien de plus simple, il suffit de remplacer le nom *MesMacrosFinalisées.txt* par *StartupMacros.txt*, de déplacer ce fichier dans le répertoire *Macros* d'ImageJ et à chaque démarrage, ImageJ chargera le ou les macros présentes dans ce fichier. Dans cette distribution d'ImageJ, le fichier *StartupMacros.txt* est utilisé. Il convient dans ce cas pour ne pas perdre les macros propres à la distribution d'éditer ce fichier et d'y ajouter à la suite les macros de son choix.

Associer une touche du clavier à une macro

Ceci est particulièrement pratique lorsque l'on utilise fréquemment une ou plusieurs macros..

Ainsi, déclarée d'une certaine manière et installée cette macro sera lancée chaque fois que l'on pressera une certaine touche, généralement F1, F2... etc

La syntaxe est la suivante:

à la place de la ligne :

```
macro "MaMacroPréfére1" {
```

il faudra utiliser la ligne :

```
macro "Macro 1 [ToucheDeVotreChoix]" {
```

Identification des fenêtres

Lorsque les macros se compliquent il faut faire face à une difficulté supplémentaire qui est la gestion de l'identité des fenêtres. Dans notre premier exemple, la de sauvegarde de projections maximale d'intensité était effectuée par l'instruction:

```
run("Tiff...", "save=/home/tristan/Desktop/Essai/Resultats/Demo1.tif");
```

Or, au moment où cette instruction est traitée, deux images sont ouvertes: la projection, mais aussi le stack utilisé pour faire la projection.

Comment imageJ sait que l'image à sauver est la projection ?

Dans ce cas précis, la réponse est simple, la projection est l'image active car elle est le résultat de la dernière opération effectuée..

Comment faire si vous désirez copier l'image, la convertir en 8 bit et sauver une version 8 bit et une 16 bits..

Il existe deux possibilités.. la première consiste à utiliser la séquence suivante:

ouvrir le stack;

faire la projection;

sauver la projection en 16 bits;

dupliquer la projection;

convertir la copie de la projection en 8 bits;

sauver la copie de la projection en 8 bits;

fermer la copie de la projection;

fermer la projection ;

fermer le stack;

Dans cet exemple, vous vous arrangez pour que la fenêtre active soit toujours celle dont vous avez besoin... cette méthode est utilisable, mais nécessite d'être très attentif et devient difficilement applicable dans les programmes plus compliqués. Par ailleurs, une intervention extérieure pourrait entraîner une modification de la fenêtre active non voulue et faire planter la macro.

Le plus sûr et le plus rigoureux est d'appeler les fenêtres par leur identité.

Pour cela, nous allons utiliser la fonction `getImageID()` de la bibliothèque de fonction ImageJ (Build-in Macros functions).

Cette fonction retourne l'identifiant de la fenêtre active, cet identifiant est un nombre négatif unique.

La fonction `selectImage(identifiant)` permet d'activer la fenêtre dont l'identifiant est passé comme argument.

Ainsi, la nouvelle séquence des événements pourrait être:

ouvrir le stack;

```
identifiant stack=getImageID();  
faire la projection;  
identifiant projection =getImageID();  
selectImage(stack);  
fermer le stack;  
selectImage(projection)  
dupliquer la projection;  
identifiant copie_projection = getImageID();  
convertir la copie de la projection en 8 bits;  
selectImage(projection);  
sauver la projection;  
fermer la projection ;  
selectImage(copie_projection);  
sauver la copie de la projection;  
fermer la copie de la projection;
```

Dans cet exemple, on passe d'une fenêtre à l'autre en utilisant les identifiants des images.

Le code est plus lourd, mais l'exécution plus rapide (puisque l'on peut fermer le stack dès qu'on a fait la projection) et plus propre puisque chacune des fenêtres est appelée par son identifiant..

exemple:

```
for (i=1; i<51; i++){  
a=Rename(i);  
open("/home/tristan/Desktop/Essai/Demo"+a+".stk")  
stack = getImageID();  
run("Z Project...", "start=1 stop=25 projection='Max Intensity'");  
projection= getImageID();  
selectImage(stack);  
  
selectImage(projection);
```

```
run("Duplicate...", "title=copy_projection");
copieprojection=getImageID();
run("8-bit");
selectImage(projection);
run("Tiff...",
"save=/home/tristan/Desktop/Essai/Resultats/16_bits_Demo0"+a+".tif");
close();
selectImage(copieprojection);
run("Tiff...", "save=/home/tristan/Desktop/Essai/Resultats/8_bit_Demo0"+a+".tif");
close();
}
```

```
function Rename (i) {
if (i<10) a= 0+"i";
if (i>=10) a="i";
return a;
}
```

Cette macro permet de fermer le stack avant de travailler sur les projections.. par ailleurs, l'identification des fenêtres rend en général le code plus robuste..

Pour information, il existe dans la même famille deux autre fonctions `isOpen(id)` et `isActive(id)` permettant de déterminer respectivement, si une fenêtre est ouverte, ou est active.

Il existe aussi une autre famille de fonctions `getTitle()/selectWindow("name")`, permettant de passer d'une image à l'autre en les sélectionnant par leur nom. Elles sont plus convivial et permettent d'obtenir un déroulement de programme moins rapide, surtout lorsque de nombreuses fenêtres sont ouvertes.

Pour plus d'information : <http://rsb.info.nih.gov/ij/developer/macro/functions.html>.

Communication avec l'utilisateur

Vous serez nécessairement amenés à afficher différents de message à l'écran. Soit pendant la phase de développement de votre programme pour tester et vérifier différentes points, soit, dans la version définitive de votre programme, pour

communiquer avec l'utilisateur.

Différentes fonctions sont à votre disposition dans la bibliothèque de fonctions Macro d'ImageJ pour vous permettre de subvenir à ces besoins.

print

Syntaxe : `print(string)`, écrit dans la fenêtre log la chaîne de caractère entre parenthèses.

showMessage

Syntaxe : `showMessage(["titre,]"message")`, affiche "message" dans une boîte de dialogues. Le titre est optionnel (d'où les crochets).

il existe une petite variante avec un bouton Cancel.. cliquez sur le bouton arrête l'exécution de la macro.. syntaxe: `showMessageWithCancel(["titre",]"message")`

write

Syntaxe : `write(string)`, identique à `print(string)` mais la sortie s'effectue vers la fenêtre Results au lieu de log

getString

Syntaxe : `getString("invite", "valeur par défaut")`, affiche une boîte de dialogues et renvoie la saisie de l'utilisateur. Le premier argument est l'invite, le second la valeur d'initialisation de la saisie.

Une saisie nulle ou le fait de cliquer sur « Cancel » arrête le déroulement de la macro..

exemple:

```
a=" quel est ton nom...?";
```

```
b="Blabla";
```

```
b=getString(a, b);
```

Ouvrira une boîte de dialogue avec comme invite : "quel est ton nom...? " et comme suggestion de réponse : Blabla

Le nom saisi par l'utilisateur sera affecté à la variable b, qui n'a donc plus la valeur : Blabla... ce que l'on peut vérifier en faisant un `showMessage("Votre nom est ", b)`

getResults

syntaxe: `getResult("Colonne", ligne)`

Récupère une mesure de la fenêtre Results d'ImageJ. Le premier argument doit correspondre à l'intitulé d'une colonne.. (Area, Mean, Max...etc). Le deuxième argument indique le rang de la mesure.. (première mesure, deuxième mesure..). La

première mesure correspond au rang 0 (comme pour les tableaux..) la dernière au rang nResults, variable prédéfinie d'imageJ (égale au nombre de mesure moins 1). Depuis la version 1.34g d'imageJ il est possible d'omettre le deuxième argument, sa valeur par défaut étant alors égale à nResults-1.

Incorporer des commandes et des plugins dans vos macros : les limites

L'incorporation des commandes et plugins dans les macros n'est pas toujours chose facile..

En effet, pour de nombreux commandes et plugins, l'interfaçage ou la passation d'arguments n'a pas été prévue, rendant leur utilisation dans les macros difficile, voir impossible..

Transformation d'une image en tableau

Dans de nombreux cas, il est nécessaire de transformer l'image en tableau de valeurs pour pouvoir travailler plus facilement et générer des fonctions inexistantes

Les tableaux dans imageJ sont des tableaux une dimension.. l'image sera donc stockée sous la forme d'une seule ligne..

La liste de commandes classiquement utilisée pour réaliser un tel exploit est la suivante:

```
largeur=getWidth();
hauteur=getHeight();
nombreDePixels=largeur*hauteur;
MatriceImage=newArray(nombreDePixels);

for (y=0; y<(hauteur); y++){
for (x=0; x<(largeur); x++){
MatriceImage[(largeur*y+x)]=getPixel(x, y);
}
}
```

A partir du tableaux MatriceImage, vous allez pouvoir effectuer de nombreuses opérations..

Par exemple, si vous souhaitez modifier l'histogramme d'une image afin qu'il soit étalé

de 0 à 255, il suffit de trouver le min et le max de l'image puis d'effectuer pour chacun des pixels la correction suivante: Intensité normalisée = $(I - I_{\min}) / (I_{\max} - I_{\min}) * 255$.

Pour trouver le min et le max :

```
Imax=0;
Imin=255;
for (i=0; i<nombreDePixels; i++){
Imax=maxOf(Imax, MatriceImage[i]);
Imin=minOf(Imin, MatriceImage[i]);
}
```

Reste à faire le calcul de modification d'intensité des pixels:

```
NouvelleMatriceImage=newArray(nombreDePixels);
for (i=0; i<nombreDePixels; i++){
NouvelleMatriceImage[i]=((MatriceImage[i]-Imin)/(Imax-Imin)*255);
}
```

Puis à remplacer les valeurs de pixels dans l'image d'origine.. en utilisant la fonction setPixel(x, y, value) qui permet l'affectation d'une valeur d'intensité eu pixel x,y.

```
for (y=0; y<(hauteur); y++){
for (x=0; x<(largeur); x++){
setPixel(x, y, NouvelleMatriceImage[(largeur*y+x)]);
}
}
```

L'affichage de l'image est automatiquement mis à jour... l'image apparaît avec un affichage utilisant réellement les 255 niveaux disponibles en 8 bits.

Gagner du temps avec le mode Batch

syntax: `setBatchMode(boolean)`

Lorsque le booléen a comme valeur « True » (`setBatchMode(true)`), alors il n'y a pas de représentation des images à l'écran..

Vous êtes en mode batch.. les vitesses de calcul sont beaucoup plus rapides..

Si vous voulez revenir à un mode avec représentation des images à l'écran, sortez du

mode batch en utilisant la commande : `setBatchMode(false)`.

Conclusion

En quelques pages, nous vous avons présentés quelques unes des nombreuses possibilités que présente ImageJ en programmation macro.

Il existe dans la bibliothèques de fonctions macro d'ImageJ de très nombreuses fonctions, décrites de façon explicite et de nombreux exemples permettant de saisir toutes les subtilités de la programmation macro d'ImageJ (<http://rsb.info.nih.gov/ij/developer/macro/functions.html>). Le nombre de ces fonctions est en croissance permanente et il est donc de plus en plus simple de trouver une solution en programmation macros, à de nombreuses problématiques rencontrées quotidiennement...

Toutes les corrections, critiques, améliorations et ajouts sont les bienvenus. N'hésitez pas à nous contacter..(chamot@ijm.jussieu.fr, piolot@ijm.jussieu.fr)