

Partie 3

Principes du traitement d'images
Champs d'applications, rappel d'optique
Algorithmes et notions mathématiques

Chapitre 1

Principaux champs d'application

I. Les domaines où l'image intervient

L'image est impliquée dans de nombreux domaines. On peut les différencier selon ce qu'on a au départ, et ce qu'on veut. En Computer Graphics, on construit un modèle et on le projette pour obtenir une image. En Computer Vision, on analyse l'image pour avoir des informations.

Input / Output	Image	Description
Image	Image processing	Computer vision
Description	Computer graphics	Artificial Intelligence

Concernant les grandes applications actuelles, on peut en citer quatre :

- Reconstruction d'un objet d'après plusieurs images.
- Pattern recognition, comparaison d'objets, OCR (reconnaissance de caractères). Lorsqu'on fait une comparaison, comme les modèles peuvent être à des tailles différentes sur des images de couleurs différentes, on a besoin d'une représentation plus indépendante, comme des statistiques.
- Calcul de mouvements entre des séries d'images.
- Trouver une image dans une base de données comme le net. Attention : la question que l'on soumet n'est pas un mot mais une image ; si je veux trouver un cheval, j'en donne un exemple plutôt que de me forcer à le décrire !

Il n'y a pas de frontières bien claires entre les domaines. Parfois, le traitement d'image se mêle à l'analyse d'images, ou la vision informatique. Par exemple, l'analyse automatique des textes (l'OCR) fait appel aux deux : l'acquisition, le pré-processing de l'image, l'extraction des caractères et leur description relève du traitement d'images ; donner un sens au texte complet est plutôt du côté de la vision informatique.

II. Vers une définition du traitement d'images

Une image peut-être définie comme une fonction en deux dimensions $f(x,y)$ où x et y sont les coordonnées spatiales, et la valeur prise par f (son amplitude) est appelée l'intensité ou niveau de gris. Quand x , y et les valeurs prises par la fonction sont des quantités discrètes, finies, alors on a une image digitale. Le traitement d'images digitales sous-entend que le traitement est effectué par informatique.

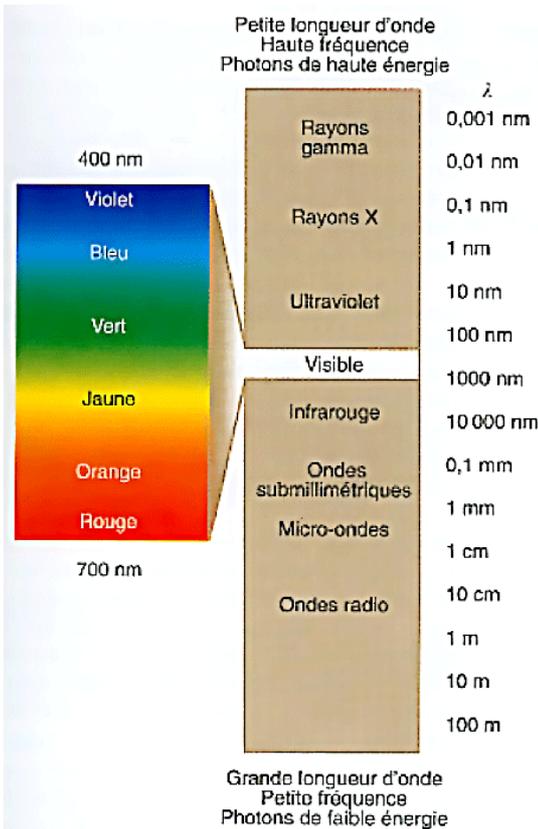
Cependant, traiter une image uniquement si on la voit uniquement comme pixels n'est pas évident. On peut appliquer des opérations dessus très facilement, mais comment s'assurer d'avoir le résultat voulu ? On préfère considérer l'image sous forme de signal : alors on sait quel traitement appliquer au signal, et on conçoit le filtre. Le passage d'une image spatiale à un signal est assuré par les séries de Fourier (si on veut traiter toute l'image), ou les *wavelets* (pour un traitement local) ; en théorie on utilise la DFT (*Discrete Fourier Transform*), mais on se contente généralement de la FFT (*Fast Fourier Transform*).

III. Les origines

Les premières applications du traitement d'image furent l'industrie du journal, où les images étaient envoyées par câble sous-marin entre Londres et New York. L'introduction du système Bartlane dans les années 1920 a réduit le temps de transport d'une image : de plus d'une semaine à moins de trois heures ! Des équipements d'impression spécialisés codaient l'image pour une transmission câblée, et elle était reconstruite de l'autre côté. Certains des problèmes pour améliorer la qualité de ces images étaient reliés aux procédures employées et à la distribution des niveaux d'intensité. Les premiers systèmes Bartlane ne pouvaient coder que sur 5 niveaux de gris : en 1929, on passe à 15 niveaux.

IV. Premières utilisations d'images digitales

Les premiers ordinateurs assez performants pour faire du traitement d'images apparurent dans les années 1960. La nécessité du traitement d'images digitales est intimement lié à la disponibilité de ces machines et le lancement du programme spatial américain. Au laboratoire de Pasadena (Californie), on a travaillé en 1964 pour rectifier les distorsions des images de la lune transmises par Ranger 7. Cette expérience a servi de base pour continuer de faire évoluer les méthodes d'amélioration et de restauration d'images, utilisées pour les missions Surveyor (lune), Apollo, etc...



En parallèle avec les applications spatiales, on a commencé à utiliser ces techniques dans les années 1970, pour l'imagerie médicale, les observations de la terre et l'astronomie. En particulier, signalons l'invention de la CAT (Computerized Axial Tomography), où un anneau de détecteurs encerclent un objet/patient, et où une sorte de rayons X est en rotation. Ce procédé fut inventé par Hounsfield et Cormack, prix Nobel de médecine en 1979 (les rayons X ont été découverts en 1901 par Roentgen, prix Nobel de physique).

A présent, quasiment tous les domaines font appel aux images digitales. Le moyen le plus simple de faire une étude est de classer les images en catégories selon leurs sources : visible, rayons X, ultraviolet... Les principales sources d'énergies utilisées pour l'image sont aujourd'hui dans le spectre électromagnétique, mais on peut également signaler l'acoustique, l'ultrasonique et l'électronique (microscope).

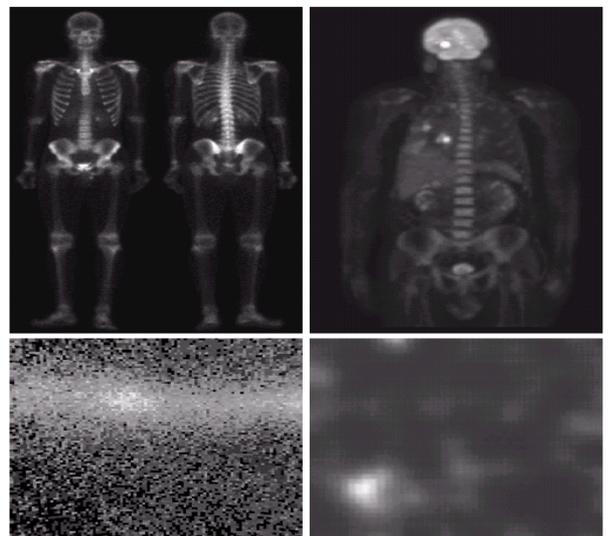
($\lambda = c_v / \nu$, où c_v est la vitesse de la lumière dans le milieu considéré pour la fréquence ν . Ondes radio et radar sont produits par les courants électriques haute fréquence, infrarouge/visible/ultraviolet résultent de transitions électroniques dans les atomes sur les électrons périphériques, rayons X proviennent de radioactivité et provoquent des transitions électroniques dans l'atome sur l'électron de cœur)

V. Images sur les rayons gamma

Les utilisations majeures sur les rayons gamma comprennent la **médecine nucléaire** et les **observations astronomiques**. On injecte au patient des isotopes radioactifs qui émettent des rayons gamma, et on collecte les émissions avec des détecteurs pour produire une image.

Une étoile dans la constellation du Cygne ayant explosé il y a 15,000 ans a généré un nuage de gaz très chaud (*Cygnus Loop*) ; en utilisant le rayonnement naturel de l'objet, on a produit l'image en bas à gauche. Celle en bas à droite vient d'un réacteur nucléaire, avec une radiation forte visible dans le coin inférieur gauche.

FIGURE 1.6
Examples of gamma-ray imaging. (a) Bone scan. (b) PET image. (c) Cygnus Loop. (d) Gamma radiation (bright spot) from a reactor valve. (Images courtesy of (a) G.E. Medical Systems, (b) Dr. Michael E. Casey, CTI PET Systems, (c) NASA, (d) Professors Zhong He and David K. Wehe, University of Michigan.)



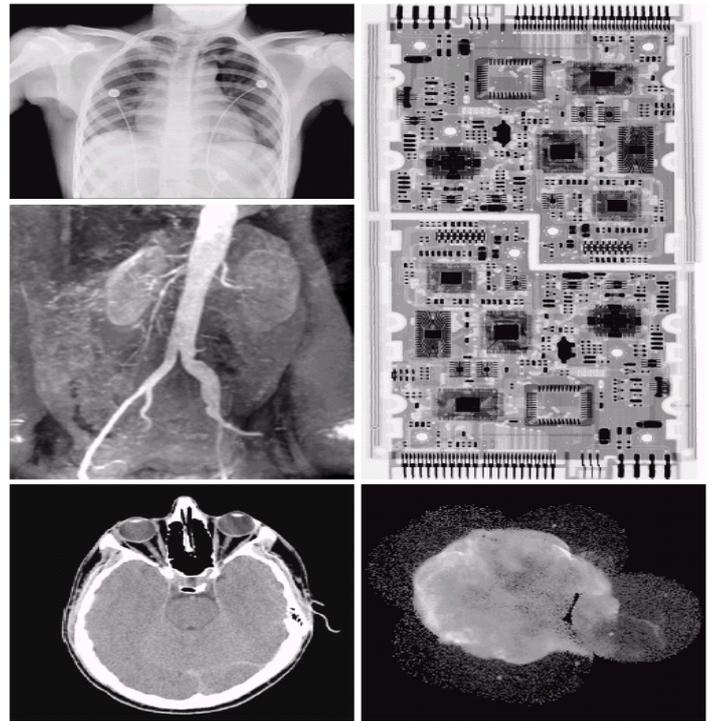
VI. Images sur les rayons X

Les rayons X sont une des plus vieilles sources du rayonnement électromagnétique utilisées en imagerie. On s'en sert pour les diagnostics, mais aussi beaucoup en industrie et astronomie.

En médecine, on utilise un tube à rayon X, qui est un tube à vide avec une cathode et une anode ; on chauffe la cathode, relâchant les électrons libres qui vont rapidement sur l'anode, et relâchent de l'énergie sous forme de rayons X lorsqu'ils heurtent le noyau.

L'angiographie (étude des vaisseaux sanguins) est une application majeure en radiographie. On injecte par un cathéter un produit augmentant le contraste des vaisseaux sanguins par rapport au reste (Iode ou CO_2), ce qui permet de bien les discerner. On utilise la différence entre l'image résultante et celle avant injection pour voir les vaisseaux.

Les CAT industriels servent à regarder s'il manque des composants ou si quelque chose est cassé dans un circuit. **Des assemblages en plastiques peuvent être pénétrés par rayons X.**



a d
b c

FIGURE 1.7 Examples of X-ray imaging. (a) Chest X-ray. (b) Aortic angiogram. (c) Head CT. (d) Circuit boards. (e) Cygnus Loop. (Images courtesy of (a) and (c) Dr. David R. Pickens, Dept. of Radiology & Radiological Sciences, Vanderbilt University Medical Center, (b) Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School, (d) Mr. Joseph E. Pascente, Lixi, Inc., and (e) NASA.)

VII. Images dans la bande de l'ultraviolet

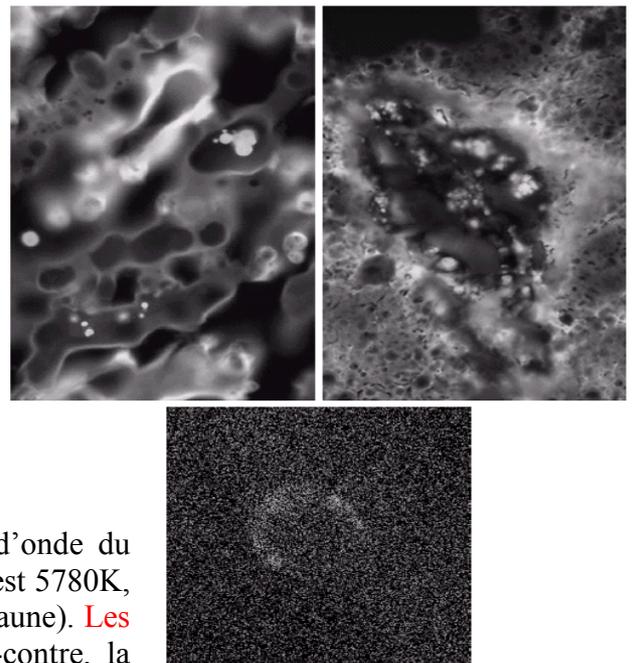
On a observé au 19^{ème} siècle que certains minéraux réagissaient quand de la lumière en ultraviolet était dirigée dessus ; le microscope à fluorescence est basé sur ce principe. Sur les exemples ci-contre, on a du blé normal, et du blé infecté. Actuellement, **l'ultraviolet est utilisé dans l'analyse des minéraux, gemmes,** et pour l'identification de toutes sortes de choses comme les billets de banque.

La loi de Wien stipule que la longueur d'onde à laquelle un corps noir (objet absorbant la lumière à toutes les longueurs d'onde) émet le plus de flux lumineux est inversement proportionnel à sa température.

Autrement dit, plus un objet est chaud, plus la longueur d'onde du rayonnement qu'il émet est courte. La température du soleil est 5780K, le maximum d'émission est donc à 500 nm, dans le visible (jaune). **Les objets très chauds émettent surtout de la lumière UV.** Ci-contre, la constellation du Cygne vue dans la région de haute énergie de la bande ultraviolet.

a b
c

FIGURE 1.8 Examples of ultraviolet imaging. (a) Normal corn. (b) Smut corn. (c) Cygnus Loop. (Images courtesy of (a) and (b) Dr. Michael W. Davidson, Florida State University, (c) NASA.)



VIII. Images dans le visible et l'infra-rouge

Sachant que la bande du visible est celle qu'on utilise le plus couramment, il n'est pas étonnant que les applications en images surpassent de loin toutes les autres en termes de portées.

On a par exemple les microscopes classiques, avec les applications pharmaceutiques ou la caractérisation de matériaux.

Il y a les emprunts digitales, que l'on modifie pour y trouver des informations, ou que l'on compare aux bases de données. On compte automatiquement les billets de banque, on lit les numéros de série pour suivre et identifier les billets... On lit aussi les plaques minéralogiques.

A température normale, les radiations émises par les objets sont dans le domaine des infrarouges (loi du rayonnement du corps noir, à nouveau). L'infrarouge excite des modes de vibration spécifiques de liaison chimiques, d'où spectroscopie.

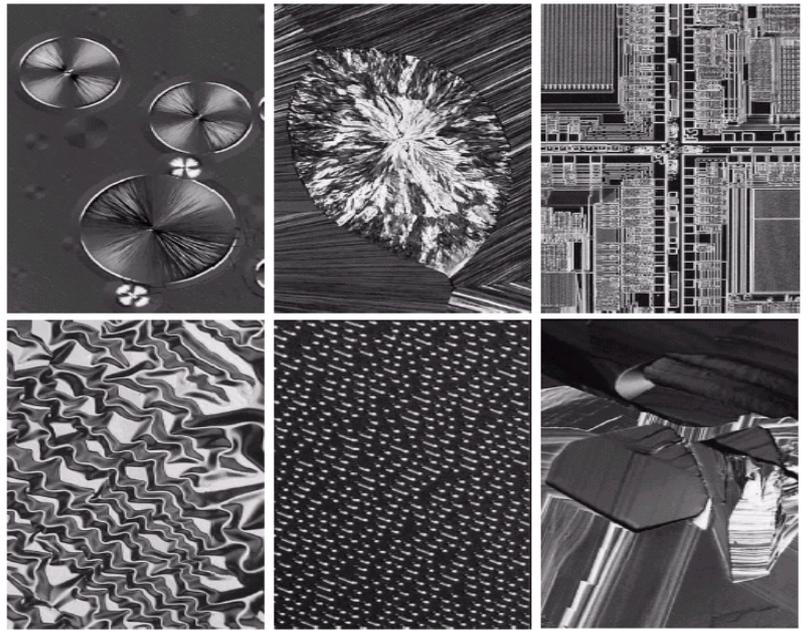


FIGURE 1.9 Examples of light microscopy images. (a) Taxol (anticancer agent), magnified 250×. (b) Cholesterol—40×. (c) Microprocessor—60×. (d) Nickel oxide thin film—600×. (e) Surface of audio CD—1750×. (f) Organic superconductor—450×. (Images courtesy of Dr. Michael W. Davidson, Florida State University.)

TABLE 1.1
Thematic bands
in NASA's
LANDSAT
satellite.

Band No.	Name	Wavelength (μm)	Characteristics and Uses
1	Visible blue	0.45–0.52	Maximum water penetration
2	Visible green	0.52–0.60	Good for measuring plant vigor
3	Visible red	0.63–0.69	Vegetation discrimination
4	Near infrared	0.76–0.90	Biomass and shoreline mapping
5	Middle infrared	1.55–1.75	Moisture content of soil and vegetation
6	Thermal infrared	10.4–12.5	Soil moisture; thermal mapping
7	Middle infrared	2.08–2.35	Mineral mapping

La télédétection (*remote sensing*) est aussi très utilisée, bien qu'elle exploite des ondes beaucoup plus larges que le visible et l'infrarouge. Le satellite LANDSAT (de la NASA) prend des images de la terre à partir de l'espace, pour surveiller les conditions environnementales de la planète. On surveille la pollution, la météo, cyclones...

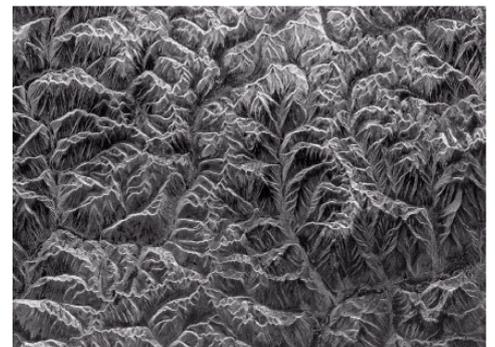
IX. Images dans les autres ondes

L'application principale dans les micro-ondes est le radar, qui sert à collecter des données à partir de régions, sans se soucier de la météo ou des conditions de luminosité ambiante. On pénètre les nuages, et selon les conditions on peut aussi voir à travers la végétation, la glace et le sable. Ainsi, le radar est souvent le seul moyen de voir les surfaces inaccessibles.

Les ondes radios sont utilisées en médecine pour l'imagerie à résonance magnétique.

On utilise le son dans l'exploration de gisements pétroliers, et les ultra-sons pour surveiller une grossesse (obstétrique).

FIGURE 1.16
Spaceborne radar
image of
mountains in
southeast Tibet.
(Courtesy of
NASA.)



Chapitre 2

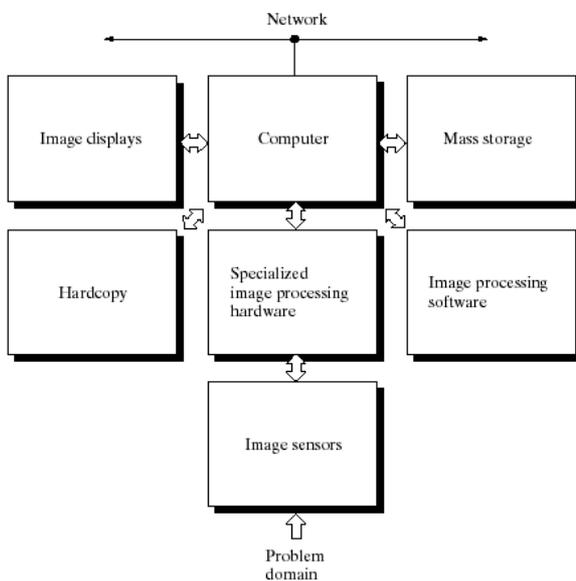
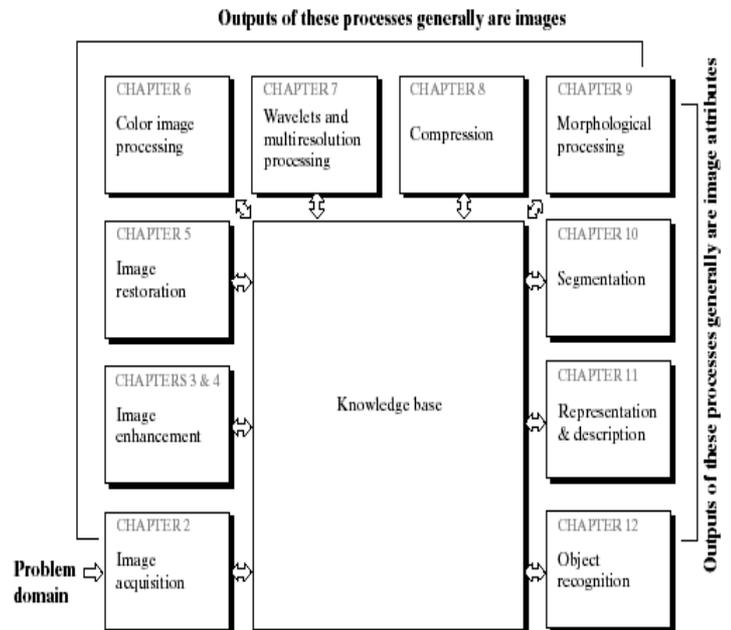
L'œil, la digitalisation et l'optique

I. Les grandes étapes du traitement d'image

◇ Acquisition de l'image

◇ Amélioration de l'image, un des aspects les plus simples et les plus attirants du traitement d'image. On veut mettre en avant certains composants de l'image, ou révéler des détails. Notons bien que l'amélioration, du point de vue humain, est purement subjective : une image qui est meilleure pour quelqu'un sera peut-être pire pour une autre personne. Par contre, du point de vue informatique, l'amélioration est objective : cela apporte des facilités sur le traitement d'informatique. On utilise les transformées de Fourier.

◇ Restauration d'image, qui traite aussi d'une amélioration sur l'apparence d'une image. Cependant, celle-ci est objective aussi pour un humain. On se base sur des modèles mathématiques ou probabilistes de la dégradation d'une image.



Il y a alors plusieurs champs, selon le but :

◇ Traitement d'images couleurs.

◇ Wavelets (ondelettes), pour la représentation d'une image dans différentes résolutions. Permet de faire de la compression de données, et des représentations pyramidales (l'image est subdivisée successivement en des régions plus petites).

◇ Compression

◇ Traitement morphologique, pour extraire les composants utiles dans la représentation et description des formes.

◇ Segmentation, partitionner l'image en objets. Une segmentation automatique est une des tâches les plus difficiles du traitement d'image ; meilleur (mais plus coûteux en temps) est ce processus, et plus grande sont les chances de faire de la reconnaissance.

◇ Représentation et description. C'est l'étape qui suit en général la segmentation ; il est à noter que selon le but, celle-ci ne peut être faite de la même manière. Ainsi, on peut choisir de représenter la donnée comme frontières ou régions complètes ; le premier choix est bon quand on s'intéresse aux caractéristiques des formes, le second est meilleur quand on s'intéresse aux propriétés internes comme les textures. Dans certaines applications, ces représentations se complètent.

◇ Reconnaissance. C'est le processus qui permet d'assigner un label à un objet, selon une description.

II. Le fonctionnement de l'œil

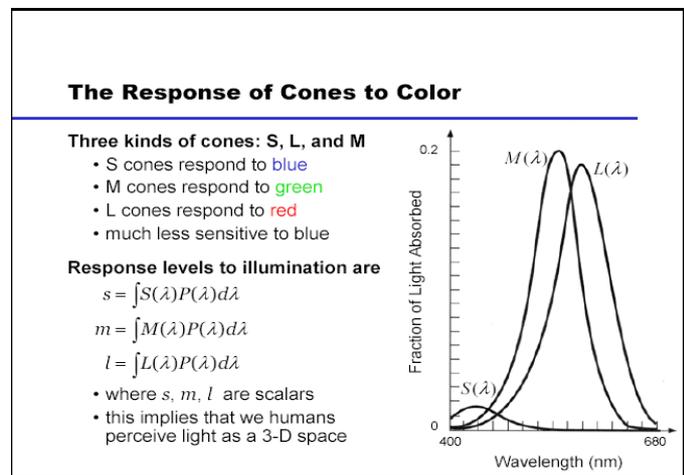
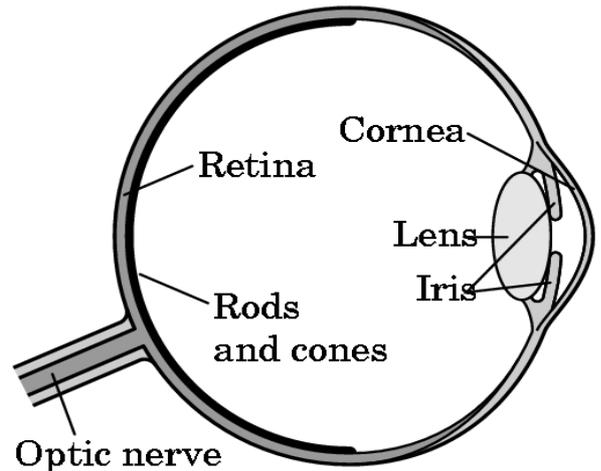
L'œil est proche d'une sphère, avec un diamètre moyenne de 20 mm. L'ouverture centrale de l'iris (la pupille) varie de 2 à 8mm de diamètre.

La rétine contient deux classes de récepteur : les cônes et les bâtonnets. Les cônes sont entre 6 et 7 millions pour chaque œil, se trouvant principalement au centre de la rétine, nommée la *fovea*, et sont sensibles à la couleur ; chaque cône dispose de son propre nerf.

Il y a de 75 à 150 millions de bâtonnets sur la surface rétinienne, plusieurs étant connectés à un seul nerf. Ils donnent une 'impression général', ne sont pas sensibles aux couleurs et sont sensibles aux faibles niveaux de lumière (c'est pourquoi la nuit nous perdons les couleurs).

Pour plus de facilité, on peut considérer la *fovea* comme étant un carré de 1,5mm par 1,5mm. La densité des cônes est de 150000 éléments par mm² ; le nombre de cônes dans la région où l'acuité est la plus forte est donc 337000.

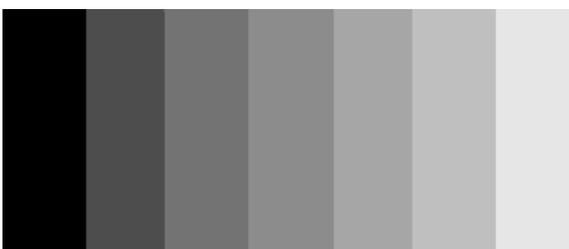
La distance entre le centre de l'œil et la rétine, dite distance focale, varie de 17 à 14 mm, au fur et à mesure sur les capacités de réfraction augmentent.



La plage de valeurs auxquelles l'œil peut s'adapter est énorme, de l'ordre de 10^{10} . L'expérience montre que la lumière *subjective* perçue est une fonction logarithmique de l'intensité lumineuse incidente sur l'œil. Il est essentiel de comprendre que l'œil ne voit pas toute sa plage de valeur simultanément ; il fait cette variation large en changeant sa sensibilité, un processus connu sous le nom d'adaptation lumineuse (*brightness adaptation*). Comparé à toute la plage, cette accessible simultanément est plutôt restreinte.

La capacité de l'œil à discerner le changement entre des intensités lumineuses est intéressante. Une expérience classique consiste à faire regarder au sujet une zone plate, illuminée de façon uniforme, assez large pour occuper entièrement le champ de vue. Cette aire est typiquement un diffuseur, comme une classe opaque, illuminé de derrière par une source d'intensité I variable. A ce champ on ajoute de la lumière d'intensité ΔI , de courte durée (flash), qui apparaît comme un cercle au centre de la zone illuminée ; l'intensité du cercle est donc $I + \Delta I$.

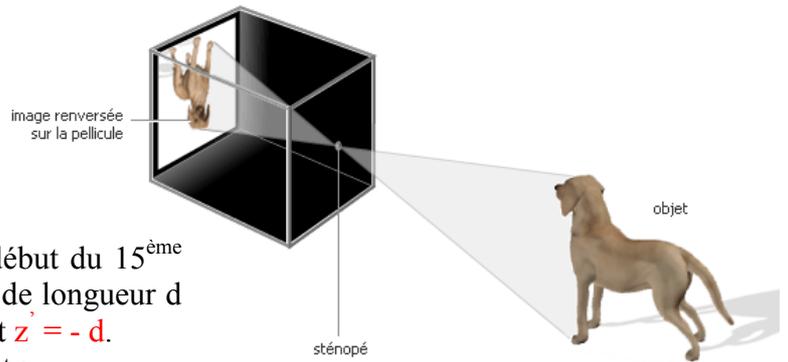
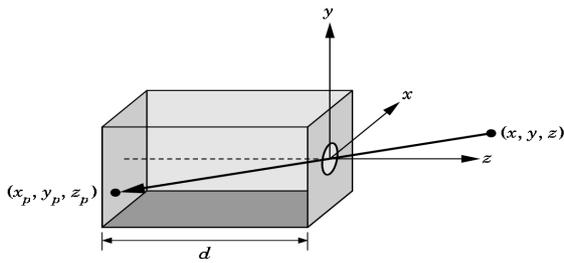
Si ΔI n'est pas assez brillant, le sujet ne le perçoit pas. La quantité $\Delta I_c/I$, où ΔI_c est l'ajout de lumière discernable 50% du temps, est nommé Ratio de Weber. Une petite valeur signifie qu'on peut discerner un petit changement d'intensité lumineuse, c'est donc bien. Au contraire, une grande valeur signifie qu'il faut qu'on ne discerne que les gros changements, ce qui est mauvais.



L'exemple ci-contre est nommé Mach bands, d'après Ernst Mach qui a décrit le phénomène suivant en 1865 : « même si l'intensité des bandes évolue de façon constante, on les perçoit comme un dégradé, ce qui signifie que notre œil fait une interpolation aux frontières. »



III. Modèle d'optique : l'appareil à sténopé



Ce modèle a été proposé par Brunelleschi au début du 15^{ème} siècle. Le point $p_{x,y,z}$ est projeté par un appareil de longueur d sur $p_{x',y',z'}$. Par construction on a immédiatement $z' = -d$.

Grâce aux triangles similaires (Thalès), on déduit :

$$-y'/d = y/z \Leftrightarrow y' = -yd/z ; \text{ de même, } x' = -xd/z.$$

Le champ, ou angle de vue, est l'angle maximal jusqu'auquel on peut prendre un objet sur la pellicule. Si h est la hauteur de notre appareil, alors l'angle est $\Theta = 2\arctan(h/2d)$, car $\tan(\Theta/2) = h/2d$.

L'appareil à sténopé 'idéal' a un champ de vue infini, ce qui signifie que chaque point est vu à un même niveau, et son image est point. En pratique, dans les optiques, les points d'un même objet ne sont pas tous projetés au même niveau ; les points non projetés sont alors flous. De même, dans les optiques l'image d'un point n'est pas nécessaire un point : nous avons plutôt un cône de vue, et l'image d'un point est un cercle ; plus le cercle est grand, et moins l'image est nette (finesse, taille du grain...).

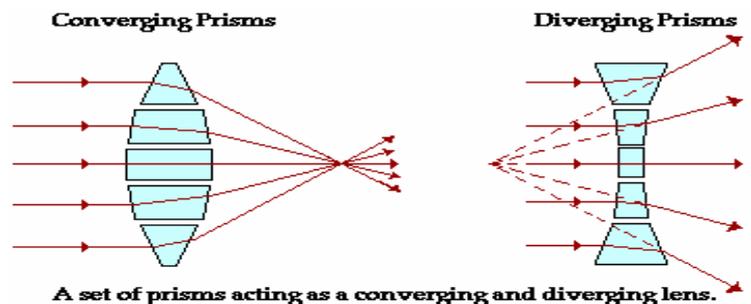
Notons également qu'en pratique, les sténopés ont de nombreux désavantages : avec un trou large, l'image est plus brillante mais le cône projeté est plus large donc on est moins net ; avec un trou réduit, on a moins de lumière...



IV. Rappels sur le fonctionnement des lentilles

En remplaçant l'appareil à sténopé par des lentilles, on résout les deux problèmes : la lentille prend de la lumière tout en gardant l'image nette, et si on choisit la distance focale appropriée alors on peut avoir n'importe quel angle de vue.

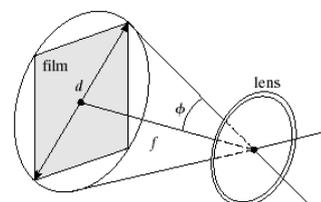
Dans le modèle idéal (*thinness*), la lentille est si fine qu'elle a une épaisseur de 0, et les rayons qui arrivent sont bien parallèles.



Le champ de vue est inversement proportionnel à la distance focale.

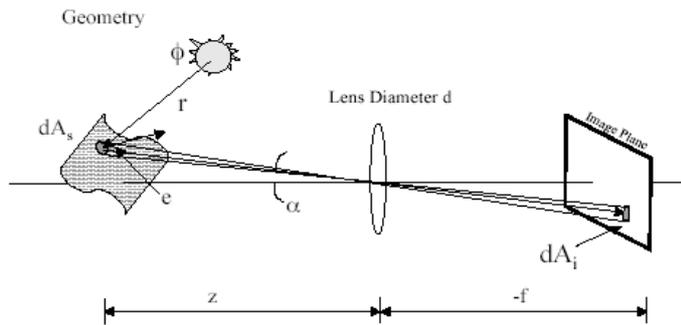
La relation exacte est $2\phi = 2\arctan(d/2f)$.

Une grande focale (*téléobjectif*) correspond à un petit angle de prise de vue ; une focale courte (*grand angle*), a une prise de vue large.



V. Éléments de Radiométrie

Goal: Relate the radiance of a surface to the irradiance in the image plane for a simple optical system.



Radiometry: Final Result

$$E_i = L_s \frac{\pi}{4} \left(\frac{d}{f}\right)^2 \cos^4 \alpha$$

Image Irradiance proportional to:

- Scene Radiance: L
- Focal Length of lens: f
- Diameter of lens: d
- Off-axis angle: α

Note: f/d is often called the "effective focal length" of the lens

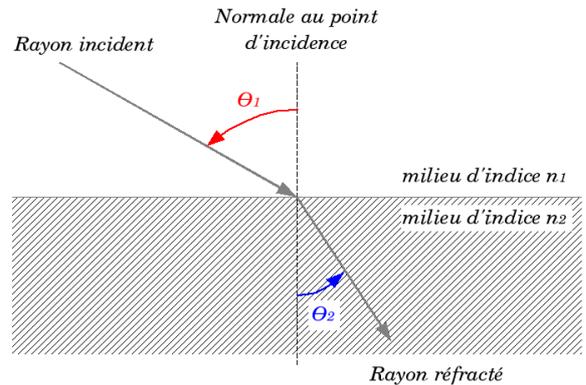
La quantité de lumière émise par une surface est nommée *scene radiance*. La quantité de lumière reçue par l'image est l'*image irradiance*. La *radiométrie* consiste à l'étude des relations entre la quantité d'image émise par une surface et celle reçue par l'image. L'*irradiance* en un point x, y est donnée par la fonction $f(x,y) = i(x,y)*r(x,y)$, où i est l'illumination et r le coefficient de réflexion.

VI. Réfraction et Réflexion

La lumière peut-être réfléchi, réfractée ou absorbée.

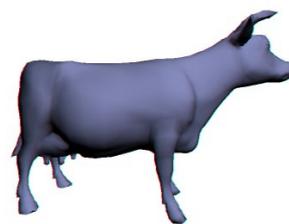
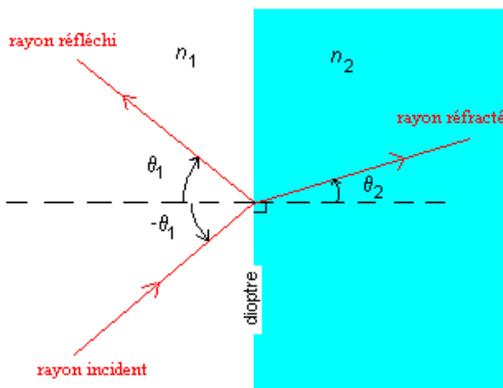
La réfraction est la déviation d'une onde lumineuse lorsqu'elle change de milieu : par exemple, de l'air à l'eau. La loi de Snell-Descartes : $n_1 \sin(\Theta_1) = n_2 \sin(\Theta_2)$.

La réflexion est le changement de direction d'une onde due à l'interface entre deux milieux ; par exemple, les rayons du soleil touchent un objet qui réfléchit les rayons déviés.



Il y a deux sortes de réflexion :

- Diffuse. Lorsque l'interface est irrégulière, la lumière est réfléchi dans plusieurs directions. C'est l'écran du cinéma : l'impact d'un seul rayon lumineux sur une surface diffusante.
- Spéculaire. Le rayon incident donne naissance à un unique rayon réfléchi. La qualité de la réflexion dépend de celle de l'interface : dès que la taille de ses défauts est du même ordre de grandeur que la longueur d'onde, l'interface tend à devenir parfaitement réfléchissante.



diffuse



spéculaire

VII. Echantillonnage pour l'acquisition d'une image

L'image est le résultat d'une fonction continue, pas discrète : il faut donc opérer un traitement afin de l'avoir sur ordinateur. Pour cela, on numérise ses coordonnées (*sampling*) ainsi que les valeurs prises (*quantization*).

Il faut déterminer trois choses :

- La résolution spatiale
- La plage de valeur prise
- Le motif utilisé pour le pixel (*tessellation pattern*) : comment distribuer le signal spatial sur le plan de l'image. Carré, triangles...
Subdivision du domaine spatial.

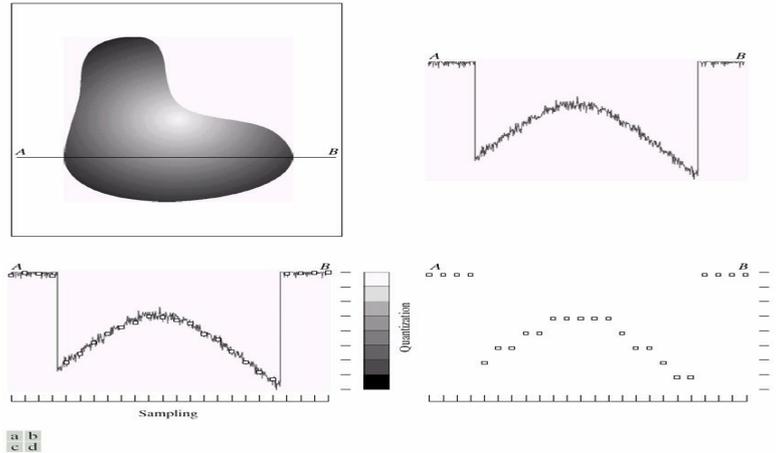
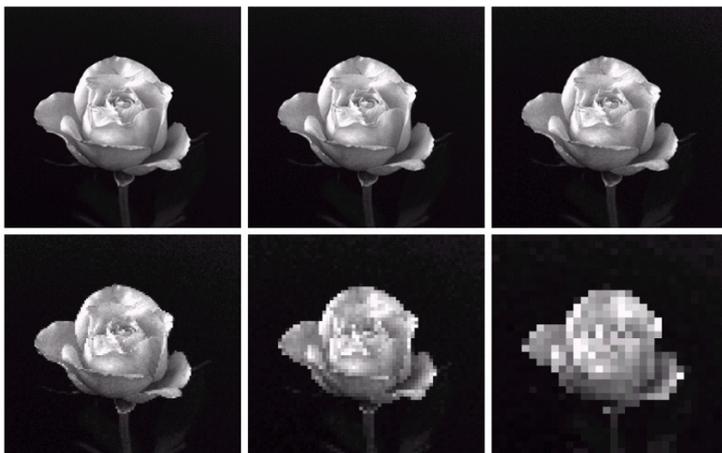


FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.



Problèmes de Sampling : *chessboard effect* / *pixellisation*

On discrétise les valeurs et on normalise pour se reporter à l'intervalle $[0, 255]$. Si on ne prend pas assez de pixels pour avoir l'image voulu, on crée des **faux contours**, comme étudié dans le Projet 1.

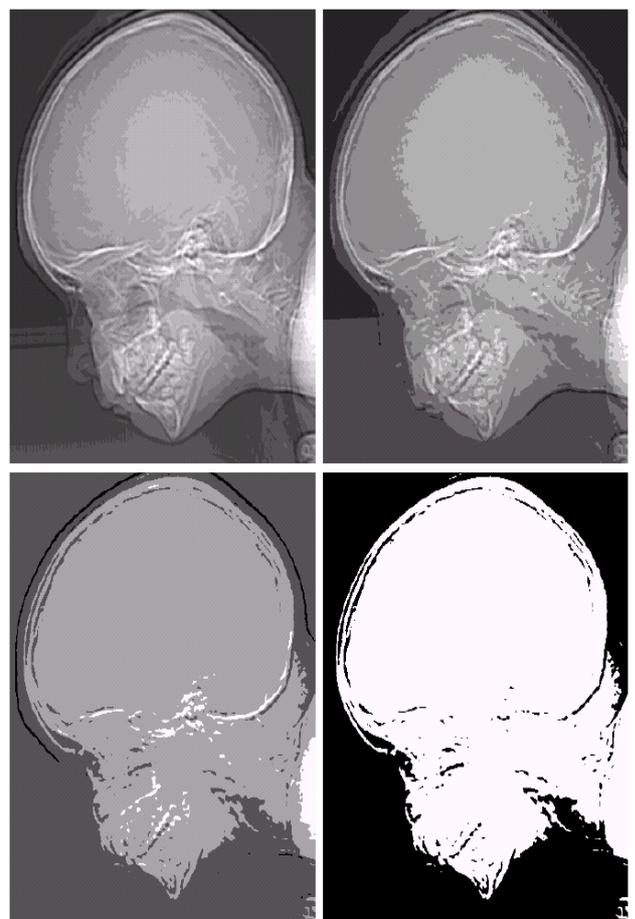
L'analyse de Fourier peut nous permettre de résoudre les problèmes de Sampling. Toute fonction en 2-dimensions $f(x,y)$ a 2 représentations : spatiale $f(x,y)$ ou comme fréquences $F(u,v)$ (superposition de toutes les fréquences qui forment f).

Pour une fonction f donnée, la transformée de Fourier nous permet d'avoir $F(u,v)$, qui est la réponse de f aux fréquences 2-dimensions (u,v) . **Le théorème (simple) de Nyquist nous dit alors : « l'échantillonnage idéal d'une fonction continue contient toutes les informations de la fonction d'origine ssi il est fait à une fréquence supérieure au double de la fréquence maximale de la fonction continue ».**

Le problème est que nous ne savons pas quelle est la plus grande fréquence présente dans l'image...

- Spatial Resolution (Sampling)
 - What should be the smallest perceivable image detail?
 - How much information of the original signal is kept?
 - Can we go back from the sampled signal to the continuous signal?
- Gray-level resolution (Quantization)
 - What should be the smallest discernible change in the gray level or color value?
 - Is there an optimal quantizer?

ef
gh
FIGURE 2.21 (Continued) (e)-(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)



Chapitre 3

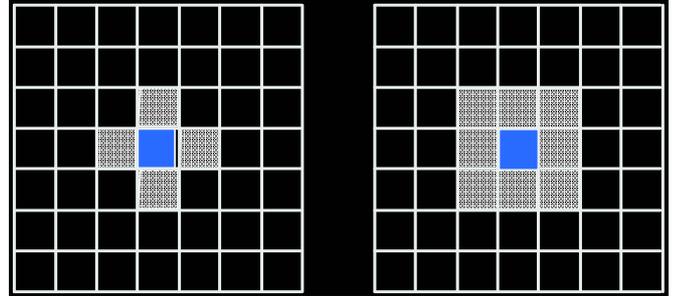
Traitement d'image en domaine spatial

I. Relations élémentaires entre les pixels

Soit p un pixel de coordonnées (x, y) . Il y a 4 voisins horizontaux et verticaux, de coordonnées $(x+1,y)$, $(x-1, y)$, $(x, y+1)$, $(x, y-1)$. On appelle ceci le 4-neighbour de p , que l'on note $N_4(p)$.

De même les 4 voisins en diagonale font $N_D(p)$.

L'union de $N_4(p)$ et $N_D(p)$ est $N_8(p)$, à savoir tous les pixels qui l'entourent immédiatement.



L'adjacence d'un pixel est considéré par rapport à un ensemble V de valeurs. Dans le premier tableau, on considère $V = \{0\}$ et $N_4(p)$. Dans le second tableau, on considère $V = \{3\}$ et $N_D(p)$. Dans le troisième, on considère $V = \{20\}$ et $N_8(p)$.

3	20	3
20	1	0
20	50	5

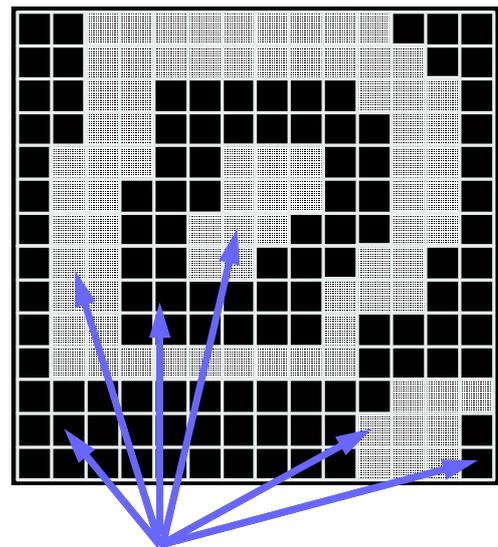
3	20	3
20	1	0
20	50	5

3	20	3
20	1	0
20	50	5

Soit S un sous-ensemble des pixels de l'image. On dit que deux pixels p et q sont connectés dans S s'il existe un chemin entre eux fait de pixels de S . Pour chaque pixel p de S , l'ensemble des pixels qui lui sont connectés dans S est nommé le *connected component* de S . S'il y a seulement un *connected component* alors S est un *connected set*.

- Pick any pixel p in the image and assign it a label
- Assign same label to any neighbor pixel adjacent to p .
- Continue labeling neighbors until no neighbors can be assigned this label
- Choose another label and another pixel not already labeled and continue
- If no more unlabeled image points, stop.

Algorithme récursif de détection des composantes connexes



6 connected components

→ Le principe dans la version itérative de l'algorithme est de scanner l'image pixel par pixel, de haut en bas et de gauche à droite, pour identifier les régions connectées. On se limite pour l'instant à des images binaires et à un modèle de δ -connectivity.

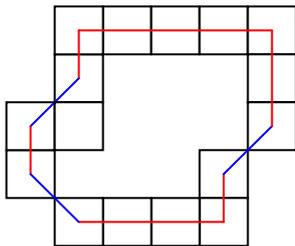
- ◇ On démarre avec le premier pixel, et on prend sa catégorie V .
- ◇ On progresse (le long de la ligne) jusqu'à arriver à un pixel p dont la catégorie est différente.
- ◇ On examine 4 voisins de p qui ont déjà été rencontrés (gauche, dessus, diagonales supérieures).
 - ◇ S'ils sont tous d'une catégorie différente de celle de p , alors on assigne à p un nouveau label
 - ◇ Si un seul des voisins est de la même catégorie que p , alors on assigne à p son label
 - ◇ Si plusieurs des voisins sont de même catégorie que p , alors on assigne à p un des labels. On met également une note stipulant l'équivalence entre labels.
- ◇ Une fois le scan fini, on rassemble les labels équivalents en classes, et on met à chaque classe un unique label. On fait alors un nouveau scan pour assigner les nouveaux labels et les couleurs.

Algorithme itératif de détection des composantes connexes

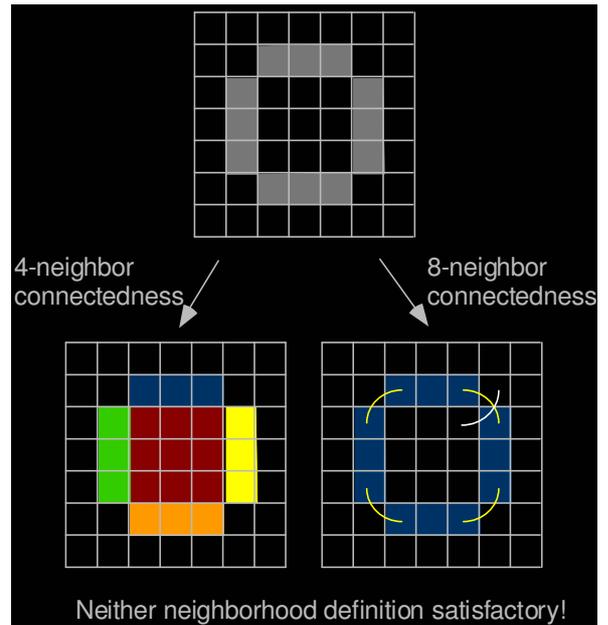
II. Séparation en régions : le problème des formes closes

Considérons ce qui se passe quand on a une surface close comme la grise ci-contre. On s'attend à ce qu'elle partitionne le plan en deux régions, mais ceci dépend du type de connectivité qu'on utilise car on est dans un cas discret ! **Il faut choisir une connectivité sur N_4 lorsqu'on traite des objets et N_8 si on s'occupe du fond ; le problème est qu'il faut déjà savoir quels pixels sont ceux de l'objet et quels sont ceux du fond...**

Lorsqu'on a un objet trouvé par une connectivité N_4 et qu'on veut compter son périmètre, on fait comme cela :



On compte chaque bloc pour 1, sauf lorsqu'on passe en diagonale auquel cas on a $\sqrt{2}$. Le total ici est $13+3\sqrt{2}$.



III. Propriétés des régions

On peut déjà calculer de nombreuses propriétés intéressantes sur les régions. Considérons une région R :

◇ L'aire est donnée par $A = \sum_{(r,c) \text{ dans } R} 1$. On fait la somme de tous les pixels, chacun comptant pour 1.

◇ Le centre (\bar{r}, \bar{c}) où $\bar{r} = (1/A) * \sum_{(r,c) \text{ dans } R} r$, et $\bar{c} = (1/A) * \sum_{(r,c) \text{ dans } R} c$.

second-order row moment:

$$\mu_{rr} = \frac{1}{A} \sum_{(r,c) \in R} (r - \bar{r})^2$$

second-order mixed moment:

$$\mu_{rc} = \frac{1}{A} \sum_{(r,c) \in R} (r - \bar{r})(c - \bar{c})$$

second-order column moment:

$$\mu_{cc} = \frac{1}{A} \sum_{(r,c) \in R} (c - \bar{c})^2$$

- common measure of circularity of a region is length of the perimeter squared divided by area. **circularity(1):**

$$C_1 = \frac{|P|^2}{A}$$

Il y a aussi un calcul classique qui s'appelle la *bounding box*, c'est-à-dire une boîte entourant l'image. Pour cela on garde trace en permanence des extrémités de la région. Si on a la boîte et son orientation, on peut s'en servir lors de la détection de pattern par exemple. Notons que ces informations sont invariantes selon l'échelle, ce qui est particulièrement utile lorsqu'on fait de la comparaison d'objets.

- Let perimeter P be the actual set of boundary pixels.
- P must be ordered in a sequence $P = \langle (r_0, c_0), \dots, (r_{K-1}, c_{K-1}) \rangle$.
- Each pair of successive pixels in P are neighbors, including the first and last pixels.

perimeter length:

$$|P| = \#\{k | (r_{k+1}, c_{k+1}) \in N_4(r_k, c_k)\} + \sqrt{2} \#\{k | (r_{k+1}, c_{k+1}) \in N_8(r_k, c_k) - N_4(r_k, c_k)\}$$

where $k+1$ is computed modulo K .

- Perimeter can vary significantly with object orientation.

- a second measure uses variation off of a circle **circularity(2):**

$$C_2 = \frac{\mu_R}{\sigma_R}$$

where μ_R and σ_R^2 are the mean and variance of the distance from the centroid of the shape to the boundary pixels (r_k, c_k) .

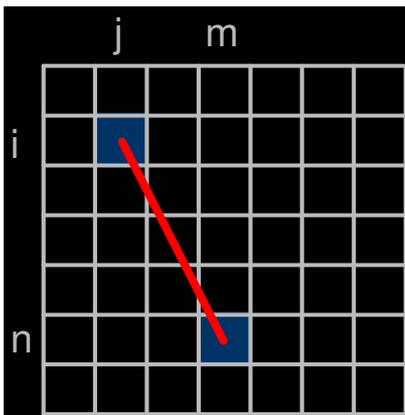
mean radial distance:

$$\mu_R = \frac{1}{K} \sum_{k=0}^{K-1} \|(r_k, c_k) - (\bar{r}, \bar{c})\|$$

variance of radial distance:

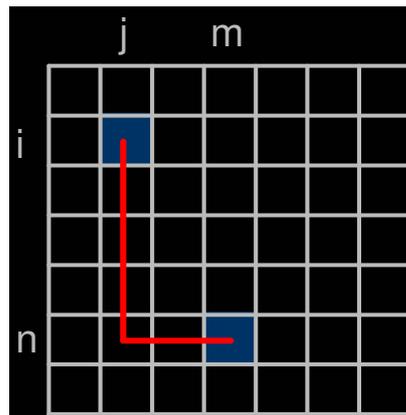
$$\sigma_R^2 = \frac{1}{K} \sum_{k=0}^{K-1} [\|(r_k, c_k) - (\bar{r}, \bar{c})\| - \mu_R]^2$$

IV. Les différentes mesures de distance



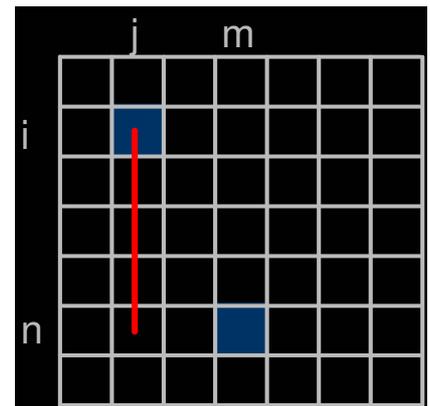
Euclidean Distance

$$= \sqrt{(i-n)^2 + (j-m)^2}$$



City Block Distance

$$= |i-n| + |j-m|$$



Chessboard Distance

$$= \max[|i-n|, |j-m|]$$

Si on représente la figure formée dans un repère selon la norme choisie, alors l'euclidien engendre un cercle, le city block (ou Manhattan) un losange, et le Chessboard un carré. Notons bien qu'une distance change selon la norme adoptée : deux pixels adjacents par la diagonale seront à une distance 1 en chessboard mais 2 en City-block (on se déplace d'un en largeur et d'un en hauteur pour la diagonale).

On dispose donc d'une définition pour le voisinage d'un point :

$$N_4(p) = \{q : d(p,q) \leq 1\} \quad \text{où } d \text{ est la city-block}$$

$$N_8(p) = \{q : D_c(p,q) \leq 1\} \quad \text{où } D_c \text{ est le Chessboard}$$

et de même D est la distance euclidienne, choisie implicitement

V. Améliorations d'une image sous forme spatiale

On considère l'image sous forme de pixels, il n'y a ici aucune transformation en signal. Le but de nos améliorations est d'avoir un meilleur résultat que le fichier original pour un certain but. Pour un humain, la qualité du résultat est hautement subjective. Pour une machine, le résultat est bon quand il permet d'amener plus rapidement ou de façon plus sûre à une conclusion.

Mathématiquement, une opération sur le domaine spatiale s'écrit sous la forme $g(x,y) = T(f(x,y))$.

◇ Lorsque $g(x,y)$ dépend uniquement de $f(x,y)$ alors T est un opérateur sur un seul point (*point-operator*).

On dit également que T est une fonction de transformation sur le niveau de gris, ou d'intensité.

On fait alors du 'point-processing' : on prend chaque point est on le modifie, il n'y a pas de transformation géométrique sur l'image. Typiquement on fait un négatif, une binarisation, ...

◇ Lorsque $g(x,y)$ dépend d'autres valeurs (telles que le voisinage du point), T est un opérateur local ou sur le voisinage (*local or neighborhood operator*).

On utilise alors un masque pour réaliser l'opération (masks, filters, kernels, templates, windows).

Un *template* est un masque particulier qui représente un objet : on l'applique partout dans l'image et on mesure la réponse pour savoir si l'objet a été détecté. Le filtrage est davantage une transformation.

L'application du masque se fait sur l'image dans sa forme spéciale. Cependant, le calcul du masque nous permettant d'arriver à un résultat bien précis est généralement fait sous forme de signaux.

◇ Si le voisinage est l'image complète, alors T est un opérateur global (*global operator*).

VI. Transformations élémentaires en *point-processing*

◇ Le négatif d'une image dont les niveaux de gris sont dans $[0, L-1]$ est donné par

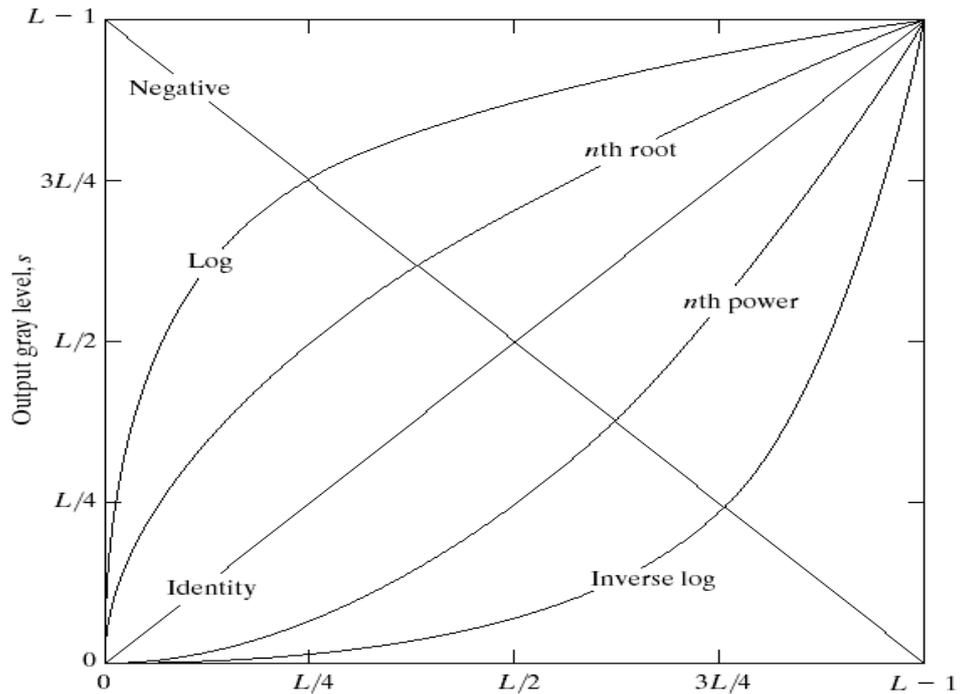
$$s = L - 1 - r$$

(s : output ; r : input)

Ceci est utile en particulier pour l'interprétation humaine, par exemple la lecture d'une mammographie. Notons que pour une image dans la plage habituelle, on a $s = 255 - r$.

◇ Transformation par le log.

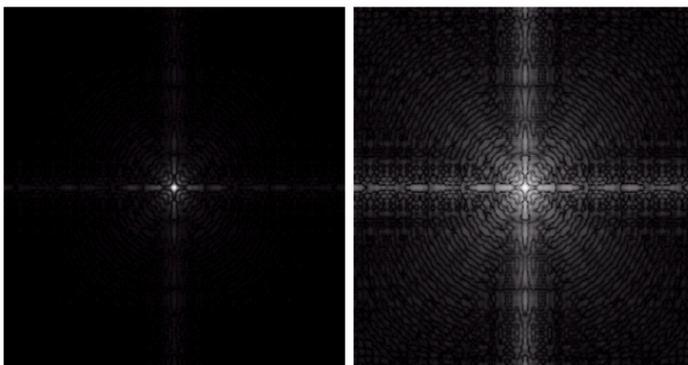
On garde l'ordre des niveaux de gris, on étend la plage dans les noirs, et on réduit le blanc. Quand on a des valeurs sur une grande plage comme une transformation de Fourier de 0 à 10^6 , alors c'est indiqué.



a b

FIGURE 3.5

(a) Fourier spectrum.
(b) Result of applying the log transformation given in Eq. (3.2-2) with $c = 1$.



Formule de transformation par le log:

$$s = c \cdot \log(1+r)$$

Il est facile de concevoir une formule de transformation logarithmique selon les buts désirés. En effet, on a :
 $r = 0 \Rightarrow s = c \cdot \log(1+r) = c \cdot \log(1) = 0$
 $R - 1 = c \cdot \log(1+L-1) \Rightarrow c = (R-1)/\log L$
 Si on regarde sur le graphe ci-dessus, on voit que les noirs sont très nettement rehaussés par le log.

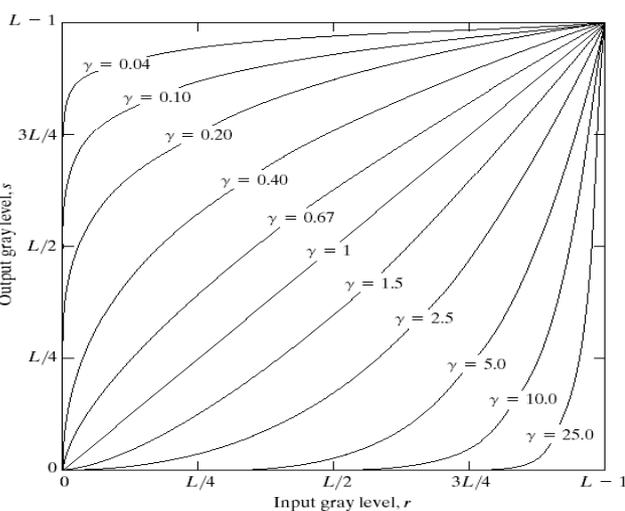


FIGURE 3.6 Plots of the equation $s = cr^\gamma$ for various values of γ ($c = 1$ in all cases).

◇ Power-law transformation.

On appelle ceci une gamma-correction. On s'en sert pour ajuster les moniteurs.

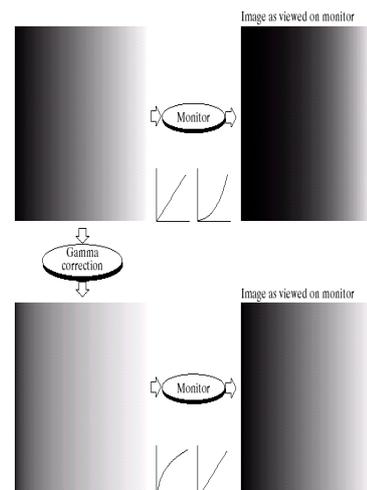
La formule est $s = cr^\gamma$. Donc, si on connaît le facteur γ d'un moniteur, on peut appliquer $(r^\gamma)^{1/\gamma} = r$. Pour être plus précis lorsqu'on fait de la calibration, on peut tenir compte de l'offset (décalage) : $s = c \cdot (r + \epsilon)^\gamma$.

(Remarquons que les effets généraux par $\gamma > 1$ sont l'opposés de ceux générés par $\gamma < 1$.)

Les moniteurs CRT ont une réponse *intensity-to-voltage* avec un γ variant de 1,8 à 2,5. En appliquant la bonne transformation inverse, on rétablit le facteur. Cela marche de même pour tous les périphériques traitant l'image ; ils ont simplement des valeurs différentes pour γ .

a b
c d

FIGURE 3.7
(a) Linear-wedge gray-scale image.
(b) Response of monitor to linear wedge.
(c) Gamma-corrected wedge.
(d) Output of monitor.



◇ Fonction définie par morceaux (piecewise-linear)

Une implémentation simple et efficace pour de nombreuses transformations peut-être de définir une fonction par morceau. En contrepartie, l'utilisateur passe plus de temps à faire la spécification, pour définir la fonction. Dans l'exemple ci-contre, on a défini une fonction par morceaux afin d'améliorer le contraste de l'image ; le but est d'augmenter la plage de valeur prise par les niveaux de gris. On peut également représenter une fonction de binarisation :

$$\begin{cases} f(x) = 0 & \text{pour } x \in [0, 125] \\ f(x) = 255 & \text{pour } x \in [126, 255] \end{cases}$$

Elle est définie en 2 parts, d'où la binarisation.

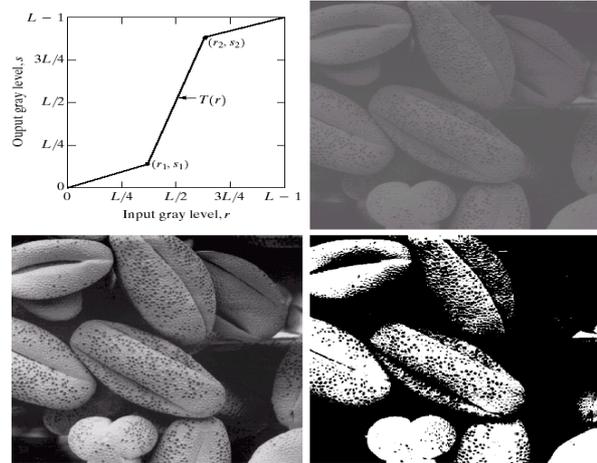


FIGURE 3.10 Contrast stretching. (a) Form of transformation function. (b) A low-contrast image. (c) Result of contrast stretching. (d) Result of thresholding. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

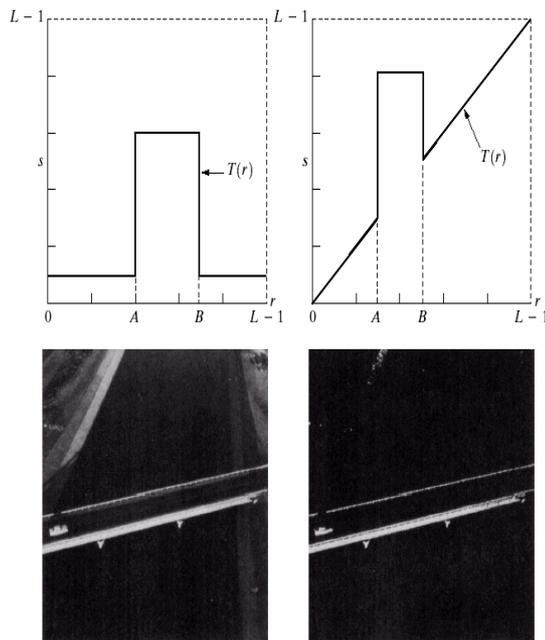
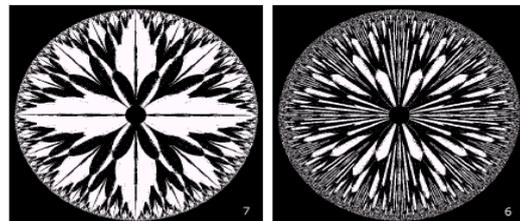


FIGURE 3.11 (a) This transformation highlights range $[A, B]$ of gray levels and reduces all others to a constant level. (b) This transformation highlights range $[A, B]$ but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

◇ Une application : grey-level slicing

On veut parfois mettre en avant seulement certains niveaux de gris d'une image, par exemple pour améliorer l'eau dans l'imagerie satellitaire ou les formes pour les rayons X. Une approche est de donner une valeur importante aux niveaux de gris qui nous intéressent, et abaisser les autres. Une autre approche est de rendre plus clair ceux qui nous intéressent, et de préserver le fond. Ces opérations s'expriment toutes avec des fonctions définies par morceaux, comme on peut le voir ci-contre.



◇ Une application : bit-slicing

Rappelons-nous que la valeur d'un niveau de gris est formée par 8 bits. On peut vouloir examiner l'information contenue dans chacun des bits de façon séparée, surtout pour la compression : quels sont les informations à garder, et où on peut-on trancher ? Comme on le voit ci-contre, le bit de poids fort contient les formes de l'image ; au fur et à mesure, on passe dans les détails.

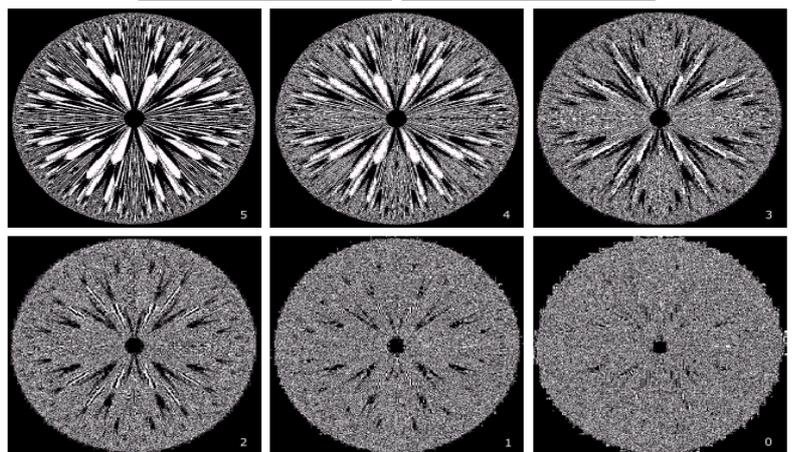


FIGURE 3.14 The eight bit planes of the image in Fig. 3.13. The number at the bottom, right of each image identifies the bit plane.

Si on veut extraire le $i^{\text{ème}}$ bit de l'image, quelle opération appliquer sur chaque pixel p ?

On cherche simplement à décaler notre bit pour le mettre en tête, et éliminer tous les autres.

◇ On décale le bit à droite pour éliminer les autres bits de droite : $s = p/2^{i-1}$.

◇ Le premier bit est celui qui nous intéresse, mais il y a encore des parasites à gauche.

Comment ne considérer que le premier bit ? Il nous dit si le nombre est pair ou impair...

On fait un modulo ! D'où la formule finale : $s_{x,y} = ((p_{x,y}/2^{8-i})\%2)*255$.

Un exemple : 1101 0000 → division par 2^5 , décalage de 5 crans : 0000 0110, et modulo * 255 : 0.

VII. Histogram Processing

0	3	3	2	5	5	
1	1	0	3	4	5	
2	2	2	4	4	4	
3	3	4	4	5	5	
3	4	5	5	6	6	
7	6	6	6	6	5	

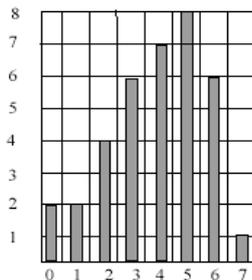
Gray Value	Count	Rel. Freq.
0	2	.05
1	2	.05
2	4	.11
3	6	.17
4	7	.20
5	8	.22
6	6	.17
7	1	.03

L'histogramme d'une image représente les fréquences relatives d'occurrences de niveaux de gris. On compte chaque niveau de gris, et on exprime le nombre de fois qu'il est présent sous forme de fréquence.

Dans le cas d'une image numérisée, c'est une fonction discrète $h(r_k) = n_k$ où r_k est le $k^{\text{ième}}$ niveau de gris, et n_k le nombre de pixels dans l'image avec ce niveau.

For an image of dimensions $M \times N$

$$\sum_{i=L_{\min}}^{L_{\max}} H(i) = (M)(N)$$



Histogram typically plots absolute count of gray levels in the image:

Chaque distribution d'histogramme peut-être spécifiée par un histogramme cible (*target*) :

- uniforme
- exponentiel (saturation du blanc)
- logarithmique (noir faible, blanc égal)
- personnalisé, selon la situation...

Prendre un histogramme et le transformer pour qu'il ressemble à la cible est l'*histogram matching*. Dans le cas où on veut qu'il soit l'histogramme uniforme (i.e. faire les niveaux automatiques...), c'est *histogram equalization*.

Soit T une transformation d'histogramme.

Soit $0 \leq r \leq 1$, le niveau de départ.

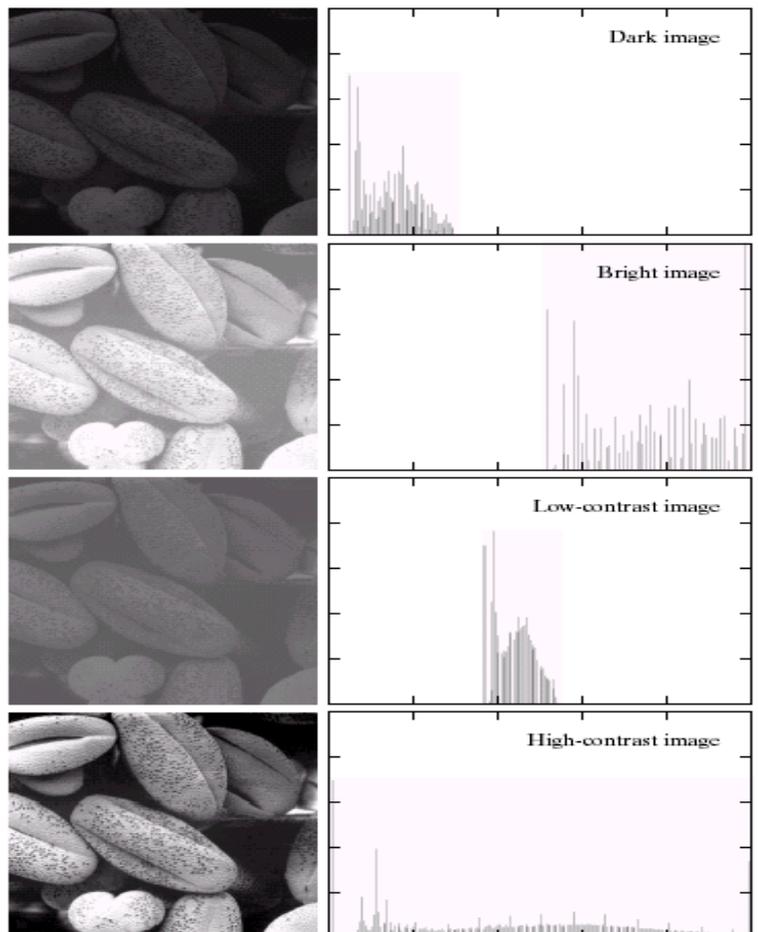
La nature formelle est $T : [0, 1] \rightarrow [0, 1]$

$$r \rightarrow s$$

Il faut la propriété $r_1 < r_2 \Rightarrow T(r_1) < T(r_2)$, autrement dit que T soit strictement croissant.

En effet, on ne veut pas changer la nature de l'image : la transformation ne doit pas modifier une partie noire en blanc... Garder la nature impose de garder l'ordre.

Il faut également que $T(r)$ soit *single-valued* : la fonction prend une valeur et une seule en r .



En résumé, T est une transformation d'histogramme si elle répond aux trois conditions suivantes :

- ◇ Strictement croissante sur $[0, 1]$
 - ◇ *Single-value* sur $[0, 1]$
 - ◇ $0 \leq T(r) \leq 1$ pour $0 \leq r \leq 1$
- } Garantie l'existence seule de la transformation inverse T^{-1}
 Mais T^{-1} n'est pas forcé d'être single-valued !

VIII. Conception de la formule d'*histogram equalization*

Rappelons quelques éléments de théorie des statistiques, utiles pour comprendre la notion d'histogramme. Soit $X : \Omega \rightarrow \mathbb{R}$ une variable aléatoire. On a $X(w) \in [a, b]$ avec une certaine probabilité.

$$P \{x \in [a, b]\} = P \{a \leq X \leq b\} = \int_a^b f(t) dt$$

$$P \{-\infty < x \leq a\} = P(X \leq a) = \int_{-\infty}^a f(t) dt = F(a)$$

Le premier est la *probabilité density function* de X (PDF), et le second la *Repartition Function* (aussi dite *Spread Function*). Appliquons ceci à la situation qui nous préoccupe : une image.

Les niveaux de gris peuvent être considérés comme une variable aléatoire dans l'intervalle $[0, 1]$. Par $p_r(t)$ on note la PDF de la variable r , représentant les niveaux de gris de l'image actuelle. L'image souhaitée après *histogram equalization* est une variable aléatoire s , où $p_s(t) = 1$ pour tout $t \in [0, 1]$; en effet, on veut une distribution uniforme, autrement dit la fonction de densité constante.

• We will design a model for histogram equalization for continuous distributions. Let $r \in [0, 1]$ represent the grey level of the image to be enhanced and s the target grey level. We look for a transformation

$$s = T(r) \quad 0 \leq r \leq 1$$

such that

1. $T(r)$ is single-valued and monotonically increasing in the interval $[0, 1]$ (why?)
 2. $0 \leq T(r) \leq 1$ for $0 \leq r \leq 1$
- The grey levels r and s may be viewed as random variables in the interval $[0, 1]$.
 - Let $p(r)$ and $q(s)$ be their corresponding probability density functions (PDF).

• We would like $q(s)$ to be the PDF of a uniform distribution over $[0, 1]$, that is

$$q(s) = 1 \quad 0 \leq s \leq 1$$

• We have

$$P\{s < T(x)\} = \int_0^{T(x)} q(t) dt = P\{r < x\} = \int_0^x p(t) dt$$

• It follows readily that

$$q(T(x))T'(x) = T'(x) = p(x)$$

• Therefore, the transformation is given by the formula

$$s = T(r) = \int_0^r p(t) dt$$

$$P \{s \leq y\} = P \{s \leq T(x)\} = P \{T(r) \leq T(x)\} = P \{r \leq x\}$$

La transformation T est bijective sur $[0, 1]$, on peut remplacer y par $T(x)$

$$\text{On utilise } T(r) \leq T(x) \Leftrightarrow T^{-1}.T(r) \leq T^{-1}.T(x) \Leftrightarrow r \leq x$$

$$= \int_0^x P_r(t) dt = \int_0^{T(x)} P_s(m) dm = \int_0^{T(x)} 1 dm = T(x)$$

On applique la fonction de répartition pour passer à l'intégrale. Comme on a une image, on a 0 au lieu de $-\infty$. 't' est une variable libre : on la remplace par m , cela n'a aucune importance, c'est une simple écriture.

• For discrete values, we deal with probabilities and summations instead of PDF and integrals. The probability of occurrence of grey level r_k is

$$p(r_k) = \frac{n_k}{K} \quad k = 0, 1, 2, \dots, L-1 \quad (1)$$

where n_k is the number of pixels with grey level r_k , K the total number of pixels in the image, and L the total number of possible grey levels in the image.

• The discrete version of the transformation T is given by the sum

$$s_k = T(r_k) = \sum_{j=0}^k \frac{n_j}{K} \quad k = 0, 1, 2, \dots, L-1 \quad (2)$$

• The output s is also assumed to have L grey levels

$$S_k = \text{floor} \left(\frac{s_k - s_0}{1 - s_0} (L - 1) + 0.5 \right) \quad k = 0, 1, 2, \dots, L-1 \quad (3)$$

$$T(x) = \int_0^x P_r(t) dt \Rightarrow s = T(r) = \int_0^r P_r(t) dt$$

La transformation T qui améliore l'image s'écrit avec des sommes puisque c'est un cas discret. Rappelons que :

- ◊ $r \in \{0, 1, \dots, L-1\}$, un pixel avec sa valeur.
- ◊ On a donc r_k pour k de 0 à $L-1$, l'ensemble des pixels pour une valeur donnée.
- ◊ $p(r_k) = n_k/K$, le nombre de pixels qui prennent le niveau de gris r_k divisé par le total des pixels.

$$s_k = \sum_{j=0}^k \frac{n_j}{K}, \quad k = 0, 1, \dots, L-1$$

IX. Discussions sur les techniques d'histogramme

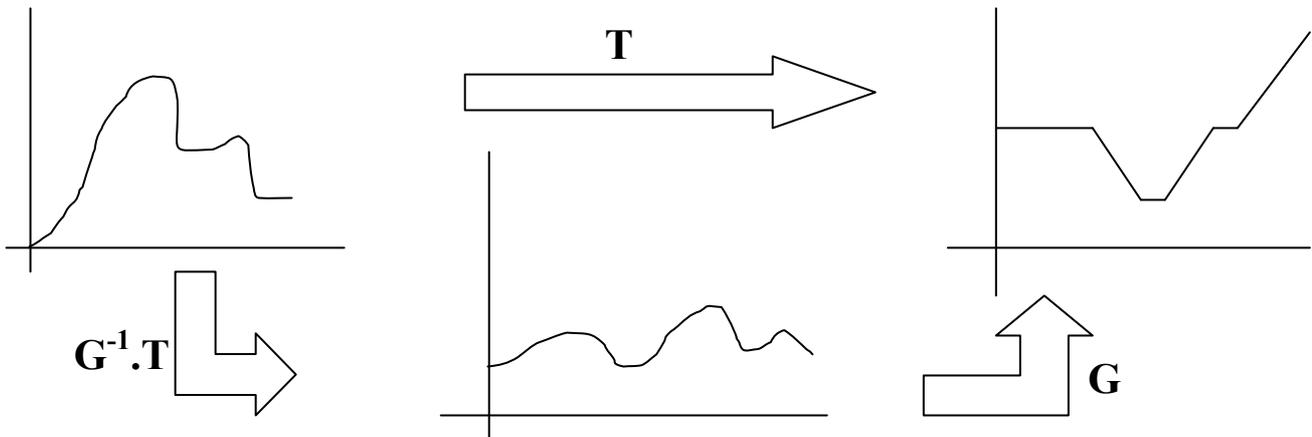
```

void HistogramEqualization(image In, image Out){
    rpt.addEvent("Histogram Equalization");
    In.histogramClean();
    In.histogramCompute();
    float s[256];
    int L = 256; // We consider an image of 256 possible grey levels
    int K = In.getwidth() * In.getheight(); // The size K of the image
    s[0] = In.histo[0] / K; // S0 = n0 / K
    for(int k = 1; k < 256 ; k++){
        int value = 0;
        for(int j = 0; j <= k; j++){
            value += In.histo[j];
        }
        s[k] = (value*255) / K;
    }
    // now, each pixel that took level Rk is mapped with level Sk
    for(int m = 0; m < In.getwidth(); m++){
        for(int n = 0; n < In.getheight(); n++){
            Out.setvalue((int)s[In.getvalue(n,m)],n,m);
        }
    }
}
    
```

La transformation vers un histogramme uniforme ne produit pas un résultat parfait comme la fonction uniforme. En effet, ce résultat ne peut-être atteint que dans un cas continue, et nous sommes en discret. Si on veut malgré tout arriver à un histogramme uniforme, il faut prendre des pixels d'une catégorie et les faire passer dans une autre ; on altère les propriétés de l'image, et la complexité augmente démesurément.

Gray Value	Actual Count	Desired Count	How to Get it.
0	0	256	22 from 7, 234 from 8
1	0	256	1 from 8, 255 from 9
2	0	256	195 from 9, 61 from 10
3	0	256	256 from 10
4	0	256	256 from 10
5	0	256	256 from 10
6	0	256	256 from 10
7	22	256	256 from 10
8	235	256	256 from 10
9	450	256	256 from 10
10	1920	256	67 from 10, 189 from 11
11	212	256	23 from 11, 10 from 12, 224 from 13
12	10	256	10 from 13, 246 from 14
13	233	256	256 from 14
14	672	256	170 from 14, 86 from 15
15	342	256	256 from 15

En *histogram matching*, on a un histogramme cible et on veut connaître la transformation T pour y arriver. On place notre histogramme dans une situation telle qu'on sache comment faire la dernière transformation.

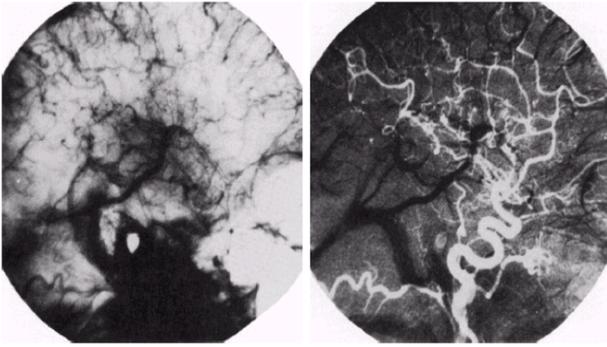


$$s = G(z) = \int_0^z P_z(m) dm \Rightarrow s_n = G(z_n) = \sum_{i=0}^n \frac{n_i}{N} \Rightarrow z_n = G^{-1}(s_n)$$

$$s_k = \sum_{j=0}^k \frac{n_j}{K} = T(r_k) \Leftrightarrow s_k = G(z_n) \Leftrightarrow G(z_n) - s_k = 0$$

Or il est impossible de trouver la valeur de z_n pour satisfaire parfaitement la dernière équation dans le cas discret. On se contente de trouver le plus petit n tel que $G(z_n) - s_k \geq 0$.

X. Différence entre deux images



a b
FIGURE 3.29
 Enhancement by image subtraction.
 (a) Mask image.
 (b) An image (taken after injection of a contrast medium into the bloodstream) with mask subtracted out.

La différence entre deux images $f(x,y)$ et $g(x,y)$ est $I(x,y) = f(x,y) - g(x,y)$.

On l'utilise principalement dans l'imagerie médicale où on l'appelle *mask mode radiography*. $g(x,y)$, le masque, est une image aux rayons X d'une région du corps du patient. $f(x,y)$ est l'image de la même région après injection d'un produit qui augmente le contraste des vaisseaux sanguins. La différence met en avant les régions où le produit est allé.

Le problème lorsqu'on prend la différence des images tel quel est qu'on se retrouve sur $[-255 ; 255]$. Une méthode est de trouver le minimum et d'ajouter son opposé ; on arrive à $[0 ; 255 + \min]$. On trouve alors le maximum et on multiplie par $255/\text{Max}$, pour arriver finalement à $[0, 255]$.

XI. Image Averaging

- Consider a noisy image

$$g(x,y) = f(x,y) + \eta(x,y) \quad (4)$$

where $f(x,y)$ is the original image and $\eta(x,y)$ is the noise distribution assumed to be uncorrelated and with average $\mu(\eta(x,y)) = 0$.

- The goal of image averaging is to reduce the noise by adding a sequence of noisy images of the same type (uncorrelated noise of same average and same variance)

$$g_i(x,y) = f(x,y) + \eta_i(x,y) \quad i = 1, 2, \dots, K$$

and taking the average image:

$$\bar{g}(x,y) = \frac{1}{K} \sum_{i=1}^K g_i(x,y) \quad (5)$$

- It follows that the expected value of $\bar{g}(x,y)$ is given by

$$E(\bar{g}(x,y)) = f(x,y) \quad (6)$$

and its variance $\sigma_{\bar{g}(x,y)}^2$ is given by the formula.

$$\sigma_{\bar{g}(x,y)}^2 = \frac{1}{K} \sigma_{\eta(x,y)}^2 \quad (7)$$

- Thus, the standard deviation at any point (x,y) of the average image is

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{K}} \sigma_{\eta(x,y)} \quad (8)$$

- As K increases, the variability (variance, standard deviation) of pixel values (noise) at each location (x,y) decreases. This means that $\bar{g}(x,y)$ approaches $f(x,y)$ as the number K of noisy images used in the process increases.

L'idée est de réduire le bruit d'une image en prenant une succession d'images et en faisant la moyenne, dans le cas où la distribution du bruit est indépendante. La formule 6 nous dit que l'espérance d'une série d'images dont on fait la moyenne est l'image sans aucun bruit. Comme l'espérance du bruit est 0, on a :

$$\begin{aligned} E\left(\frac{1}{K} \sum_{i=1}^k g_i(x,y)\right) &= \frac{1}{K} \sum_{i=1}^k E(g_i(x,y)) = \frac{1}{K} \sum_{i=1}^k E(f(x,y) + n_i(x,y)) \\ &= \frac{1}{K} \sum_{i=1}^k E(f(x,y)) + E(n_i) = \frac{1}{K} K(f(x,y)) = f(x,y) \end{aligned}$$

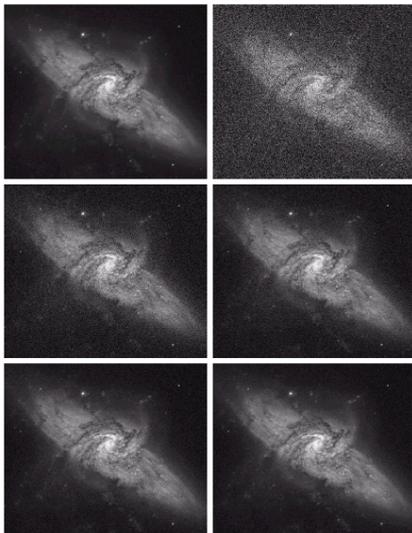
La variance (écart type, *standard deviation*) doit décroître quand K augmente.

$$Var(\bar{g}(x,y)) = E((\bar{g} - E(\bar{g}))^2) = E\left(\left(\frac{1}{K} \sum_{i=1}^k \bar{g}_i - f\right)^2\right) = E\left(\left(\frac{1}{K} \sum_{i=1}^k (\bar{g}_i - f)\right)^2\right)$$

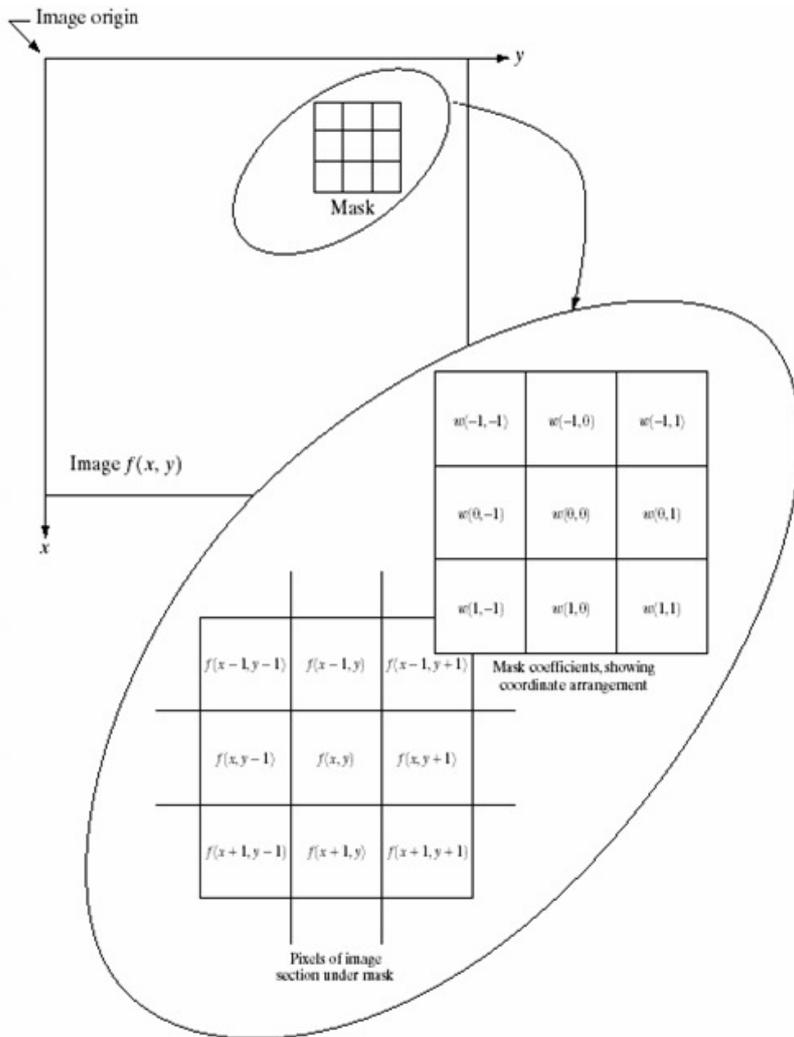
On sait que l'espérance de 'g barre' est la fonction f d'après le calcul (6). On remplace par f . Puisque f ne dépend pas de i , on le rentre dans la somme.

$$= \frac{1}{K^2} E\left(\left(\sum_{i=1}^k (g_i - f)\right)^2\right) = \frac{1}{K^2} E\left(\left(\sum_{i=1}^k n_i\right)^2\right) = \frac{1}{K^2} E\left(\sum_{i=1}^k \sum_{j=1}^k n_i n_j\right) = \frac{1}{K^2} E\left(\sum_{i=1}^k n_i^2 + \sum_{i=1}^k \sum_{i=1, i \neq j}^k n_i n_j\right)$$

$$= \frac{1}{K^2} \sum_{i=1}^k E(n_i) + \sum_{i=1}^k \sum_{i=1, i \neq j}^k E(n_i n_j) = \frac{1}{K^2} \sum_{i=1}^k E(n_i^2) \quad \text{mais} \quad E(n_i^2) = E(n^2) \forall i \Rightarrow \frac{1}{K} E(n^2)$$



XII. Concepts du filtrage spatial linéaire



Le masque ne loge pas toujours dans l'image : aux coins ou bordures il faut faire un choix. On a 2 idées :

- ◊ On ne traite pas les pixels là où cela ne loge pas
- ◊ On traite malgré tout, en allant chercher les informations nécessaires ailleurs : c'est du *padding*.
 - On met 0 là où on déborde...
 - ...ou on assume que l'image a certain propriétés cycliques et on reboucle (modulo).

On importe les masques à partir d'un fichier .mask
 # ce que fait le filtre (inclure en rapport)
 largeur (avec -1 et divisé par 2 on a le rayon)
 coeff1 coeff2 (facteur constant coeff1/coeff2)
 vall val2 val3 val4 val5...

On appelle depuis le programme, qui prend dans le répertoire mask : `Filter mean = Filter.load(« mean.ftr »)` ;
 En Scheme on pourrait faire un lambda dans le mask...
 Certains sont symétriques (comme le gaussien).

Idées pour la gestion des masques

On a un masque que l'on superpose à un certain emplacement, et qui fait toujours la même opération de type $\sum f_i w_i$ où les w_i sont les poids donnés dans le masque et les f_i les valeurs prises par l'image. Le masque, plus ou moins grand, prend en compte un rayon de points autour auquel on affecte la valeur. Par exemple avec un filtre de flou par moyenne, plus on regarde dans un rayon large, et plus on le résultat sera flou. L'écriture du masque est :

1/9	1/9	1/9
1/9	x	1/9
1/9	1/9	1/9

Le point du milieu est prit comme la somme des points environnants, divisés par 9 : on fait la moyenne. Pour ne pas perdre en précision lors des calculs, il est d'usage de donner en plus du masque un facteur qu'on applique à la fin des sommes.

- In general, linear filtering of an image f of size $M \times N$ with a filter mask of size $m \times n$ is given by the expression

$$h(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x + s, y + t) \quad (9)$$

where $m = 2a + 1$ and $n = 2b + 1$.

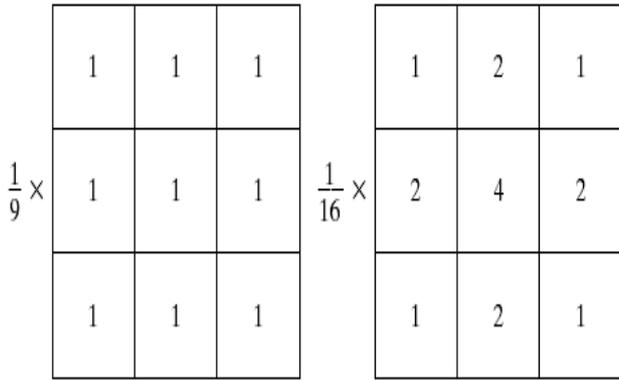
- To generate a complete filtered image, this equation must be applied for $x = 0, 1, \dots, M - 1$ and $y = 0, 1, \dots, N - 1$.
- Linear spatial filtering often is referred to as *convolving a mask with an image*.
- Similarly, filter masks are called *convolution masks*.
- In the sequel, we will justify this terminology and study convolution. But we start with an example.

En mathématiques, le produit de convolution de deux fonctions réelles ou complexes f et g se note « \otimes » ou " * " et s'écrit :

$$(f \otimes g)(x) = \int_{-\infty}^{+\infty} f(x-t) \cdot g(t) \cdot dt = \int_{-\infty}^{+\infty} f(t) \cdot g(x-t) \cdot dt$$

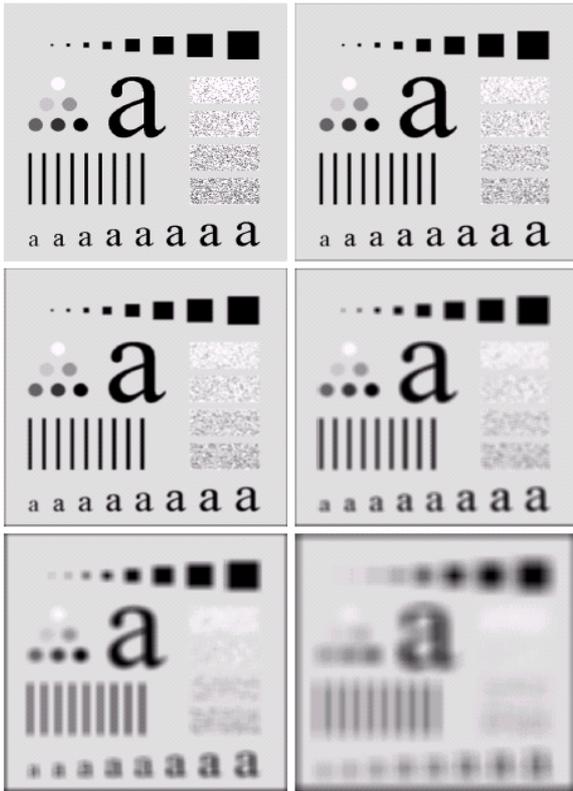
Il est très facile de montrer que le produit de convolution est distributif et commutatif. cela correspond à la multiplication des transformées de Fourier des fonctions $f \otimes g = \mathcal{F}^*(\mathcal{F}(f) \cdot \mathcal{F}(g))$ où \mathcal{F} désigne la transformation de Fourier et \mathcal{F}^* la transformation de Fourier inverse. L'intérêt principale du calcul du produit de convolution par transformées de Fourier est que ces opérations sont moins coûteuses en temps pour un ordinateur que le calcul direct de l'intégrale.

Dans la suite du cours on parlera de convolution (ou flip-inverse) lorsqu'il y aura un $-$ dans l'intégrale, et de corrélation lorsqu'il y aura un $+$.



Un des filtres spatiaux les plus classiques est l'*average*. Il consiste à prendre une moyenne pondérée des pixels. Ci-contre, le premier masque prend la moyenne avec les voisins dans un rayon de 1 pixel. Le second masque donne plus d'importance à certains pixels qu'à d'autre. La constante multiplicatrice devant chaque masque est égale à la somme de ses coefficients, puisque l'on doit faire une moyenne. On utilise deux dénominations :

- toutes les valeurs sont égales : *box filter*
- les poids sont différents : *weighted average*



Ce masque produit un effet de flou : on réduit les transitions entre les niveaux de gris. Notons que les objets de même taille que le masque sont les plus affectés par l'effet, ce qui permet par exemple de réduire le bruit. On l'appelle aussi passe-bas (*lowpass filter*) : faire la moyenne des pixels dans un voisinage local tend à supprimer les hautes-fréquences, ou adoucir l'image.

- Consider a noisy image

$$g(x, y) = f(x, y) + \eta(x, y) \quad (16)$$

where $f(x, y)$ is the original image and $\eta(x, y)$ is the noise distribution assumed to be uncorrelated and with average $\mu(\eta(x, y)) = 0$.

- The goal of the averaging mask is to produce an image with less noise

$$\bar{g}(x, y) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^a \sum_{t=-b}^b g(x+s, y+t) \quad (17)$$

- It follows that the expected value of $\bar{g}(x, y)$ is given by

$$E(\bar{g}(x, y)) = \frac{1}{(2a+1)(2b+1)} \sum_{s=-a}^a \sum_{t=-b}^b f(x+s, y+t) \quad (18)$$

- Since the variables are uncorrelated, the variance $\sigma_{\bar{g}(x,y)}^2$ is given by the formula

$$\sigma_{\bar{g}(x,y)}^2 = \frac{1}{(2a+1)^2(2b+1)^2} \sum_{s=-a}^a \sum_{t=-b}^b \sigma_{\eta(x,y)}^2$$

- Thus, the standard deviation at any point (x, y) of the average image is

$$\sigma_{\bar{g}(x,y)} = \frac{1}{\sqrt{(2a+1)(2b+1)}} \sigma_{\eta(x,y)}$$

- As a, b increase, the variability (variance, standard deviation) of pixel values (noise) at each location (x, y) decreases.
- In general, the size of the mask will depend on how important the noise is. For important noise, we use bigger masks.

Averaging Mask Theory



$$\frac{1}{N^2} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix}_{N \times N}$$

Original Image

Avg. Mask



N = 3



N = 5



N = 7



N = 11



N = 15



N = 21

- When the coefficients are not all equal to 1 in the averaging mask, we name that a *weighted averaging mask*.
- For instance, we can decide to weight the center of the mask with the highest weight and decrease the weight as a function of increasing distance from the origin.
- The basic strategy of this approach is to reduce blurring while smoothing the image.
- The general implementation for filtering an $M \times N$ image with a weighted averaging mask of size $m \times n$ is given by the expression

$$g(x, y) = \frac{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t) f(x+s, y+t)}{\sum_{s=-a}^a \sum_{t=-b}^b w(s, t)}$$

where $m = 2a + 1$ and $n = 2b + 1$.

- To generate a complete filtered image, this equation must be applied for $x = 0, 1, \dots, M-1$ and $y = 0, 1, \dots, N-1$.

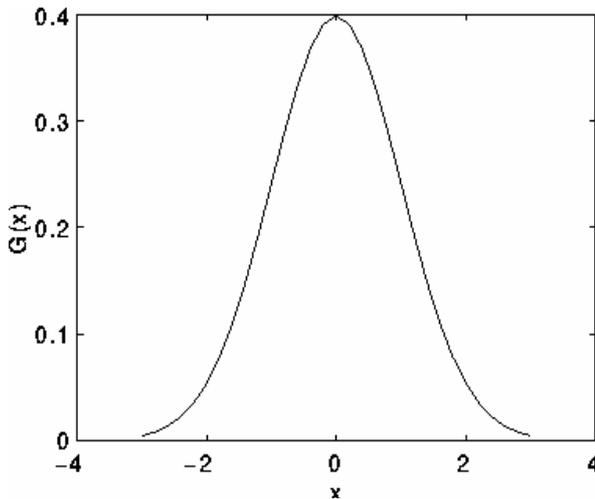
Weighted Averaging Mask Theory

On peut observer qu'une réduction de bruit en utilisant un masque moyen est la façon la plus « neutre » que l'on ait de traiter le problème : on ne connaît aucune des propriétés du bruit et on ne peut pas les exploiter. Par contre, si on connaît certaines des propriétés (sel et poivre, ...), d'autres filtres sont meilleurs, comme le médian.

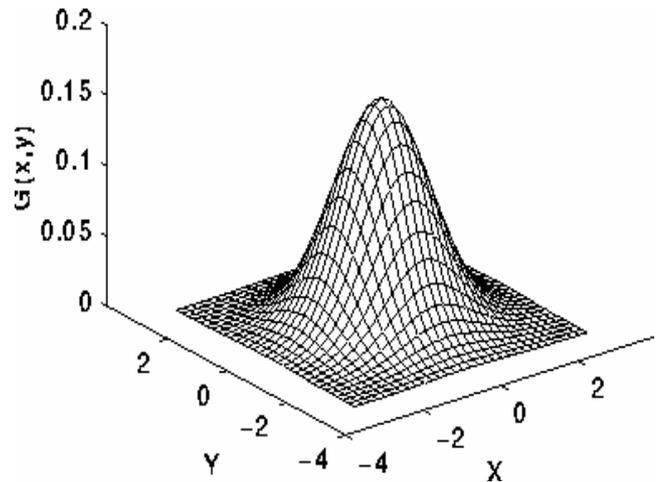
Le gaussien est un exemple de masque par moyenne pondérée. On utilise la formule suivant :

$$f(x) = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma}} \cdot e^{-\frac{(x^2 - \mu)}{2\sigma^2}}$$

La distribution correspond à ce que l'on nomme une cloche ou courbe de Gauss. μ est le centre de la cloche (l'espérance ou moyenne) et σ est l'écart-type. Il est capital de remarquer que 68% des valeurs sont dans $[\mu - \sigma ; \mu + \sigma]$, et 95% dans $[\mu - 2\sigma ; \mu + 2\sigma]$. Si on pousse jusqu'à 3σ , on a presque 100% ; comme une image est un espace fini et que la gaussienne est définie sur $]-\infty ; +\infty[$, il suffit de la translater de $+3\sigma$ pour la faire commencer à 0, et après on la coupe, ce qui nous laisse 99,99% des valeurs soit une excellente approximation.



Gaussian distribution with mean 0 and $\sigma = 1$



Gaussian distribution with mean (0,0) and $\sigma = 1$

- First, we choose the size of the mask according to the value of σ .
- Since the Gaussian is symmetric, we choose a square mask of size $m \times n$, where $m = n = 2a + 1$.
- A good (odd) integer approximation to the length 6σ would be $4 \text{ floor}(\sigma) + 1$.
- There are different strategies to compute (and approximate) the coefficients (or weights) in the Gaussian mask.
- One approach (the one you will use in your implementation) is to compute the mask weights directly from the discrete Gaussian distribution:

$$G[i, j] = c e^{-\frac{(i^2 + j^2)}{2\sigma^2}}$$

where c is a normalizing constant.

Sampling the gaussian function

Compute directly the kernel values from the gaussian function

$$g(i, j) = c e^{-\frac{(i^2 + j^2)}{2\sigma^2}}$$

1. Normalize the function and compute the values

$$\frac{g(i, j)}{c} = e^{-\frac{(i^2 + j^2)}{2\sigma^2}}$$

2. Values for a 7x7 window

(i,j)	-3	-2	-1	0	1	2	3
-3	0.011	0.039	0.082	0.105	0.082	0.039	0.011
-2	0.039	0.135	0.287	0.368	0.287	0.135	0.039
-1	0.082	0.287	0.607	0.779	0.607	0.287	0.082
0	0.105	0.368	0.779	1.000	0.779	0.368	0.105
1	0.082	0.287	0.607	0.779	0.607	0.287	0.082
2	0.039	0.135	0.287	0.368	0.287	0.135	0.039
3	0.011	0.039	0.082	0.105	0.082	0.039	0.011

3. Normalization by $g(3,3)$

(i,j)	-3	-2	-1	0	1	2	3
-3	1	3	7	9	7	3	1
-2	3	12	26	33	26	12	3
-1	7	26	55	70	55	26	7
0	9	33	70	90	70	33	9
1	7	26	55	70	55	26	7
2	3	12	26	33	26	12	3
3	1	3	7	9	7	3	1

4. Divide by $\sum_i \sum_j = 1105$

Si on fait la convolution d'un gaussien avec lui-même, alors il en résulte un gaussien avec un plus grand écart type σ . Cela signifie qu'appliquer répétitivement un petit gaussien revient à appliquer une seule fois un gros. Imaginons la situation où on doit calculer des versions de plus en plus floues d'une image pour en tirer une bonne approximation : il suffit de multiplier par le même gaussien pour passer à l'itération suivante, et on gagne en complexité. Ceci s'explique par les séries de Fourier.

Example of noise reduction



Noise-free Image

$$\frac{1}{25} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$



Zero-mean Gaussian noise, Variance = 0.01



Zero-mean Gaussian noise, Variance = 0.05



Zero-mean Gaussian noise, Variance = 0.05



Zero-mean Gaussian noise, Variance = 0.05

XIII. Eléments de filtrage spatial non linéaire

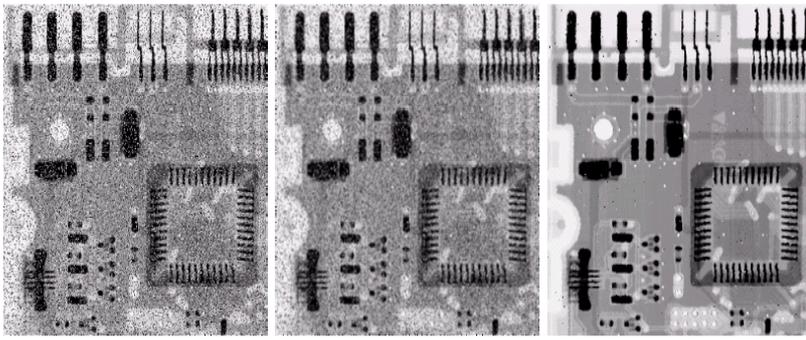
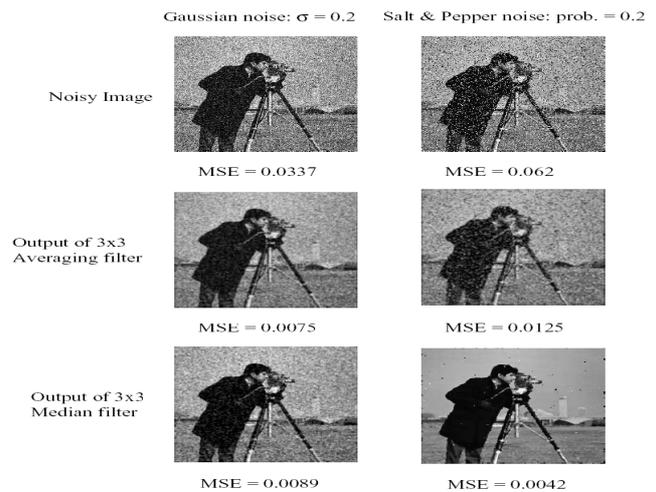


FIGURE 3.37 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3×3 averaging mask. (c) Noise reduction with a 3×3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Example of noise reduction



En filtrage non-linéaire, la réponse est basée sur un ordre des pixels contenue dans la zone du filtrage. Typiquement, on utilise des fonctions statistiques :

- ◇ filtre médian (*median filter*). On prend la valeur du milieu ; par exemple la médiane de 1 4 5 20 30 est 5. On force chaque point à être plus proche de ses voisins. Par exemple, 20 et 30 ne semblaient pas être des valeurs très cohérentes avec 1 4 5, elles pouvaient donc résulter d'une corruption d'image.
- ◇ filtre max, filtre min...

Le filtre médian est particulièrement efficace pour des bruits de type sel et poivre, et fait nettement moins de flou que les filtres linéaires de même taille. Notons que les ensembles de pixels clair ou sombre par rapport à leurs voisins sont éliminés par un filtre médian de taille $n*n$ si leur aire est moindre que $n^2/2$. Par « éliminé », on entend que leurs valeurs sont forcées à la médiane de la densité des voisins. Bien entendu, les gros ensembles de pixels sont considérablement moins affectés.

XIV. Filtres classiques

On peut utiliser les dérivées d'une image pour avoir davantage d'information. Dans la dérivée première, on regarde si elle est forte ou pas pour analyser le niveau de variation ; rappelons que les dérivées sont fortes dans les grosses transitions (sharp transition), puisque la dérivée est la plus forte là où la tangente est quasiment verticale (gros contraste entre pixels). L'utilisation de dérivées première est la méthode du gradient, que l'on peut calculer par l'algorithme de Sobel ; les dérivées secondes sont dans le Laplacien.

« En termes mathématiques, le gradient d'une fonction de deux variables (ici l'intensité en fonction des coordonnées de l'image) est un vecteur de dimension 2 dont les coordonnées sont les dérivées selon les directions horizontale et verticale. En chaque point, le gradient pointe dans la direction du plus fort changement d'intensité, et sa longueur représente le taux de variation dans cette direction. Le gradient dans une zone d'intensité constante est donc nul. Au niveau d'un contour, le gradient traverse le contour, des intensités les plus sombres aux intensités les plus claires.

L'opérateur utilise des matrices de convolution. La matrice (généralement de taille 3×3) subit une convolution avec l'image pour calculer des approximations des dérivées horizontale et verticale. Soit A l'image source, G_x et G_y deux images qui en chaque point contiennent des approximations respectivement de la dérivée horizontale et verticale de chaque point. Le calcul est alors :

$$G_x = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix} * A \quad \text{et} \quad G_y = \begin{bmatrix} +1 & +2 & +1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A$$

On observe que dans un masque pour le calcul d'une dérivée première, il y a un sens et une somme nulle.