

Traitement d'images

Polycopié de cours

Michèle Gouiffès
michele.gouiffes@universite-paris-saclay.fr

Table des matières

1 L'image : caractéristiques, représentations	4
2 Couleur	6
2.1 Synthèse additive	6
2.2 L'observateur standard de 1931	6
2.3 La représentation des couleurs RGB	7
2.4 Les espaces couleur	8
3 Amélioration d'images	10
3.1 Réhaussement de contraste	10
3.2 Réduction du bruit, lissage	11
4 Classification	14
4.1 Binarisation	14
4.2 Clustering	14
5 Régions	17
5.1 Étiquetage en composantes connexes	17
5.2 Segmentation après une classification	17
5.3 Croissance de régions	18
6 Contours	19
6.1 À partir du gradient image	19
6.2 À partir du Laplacien	20
6.3 Lissage et dérivation	21
6.4 Autres détecteurs	22
7 Morphologie mathématique	23
7.1 Dilatation et érosion	23
7.2 Ouverture et fermeture	24
7.3 Gradients morphologiques	24
7.4 Transformation en tout ou rien (<i>Hit or Miss</i>)	24
7.5 Squelette morphologique	25
7.6 Reconstruction géodésique et seuillage par hystérésis	25
7.7 Morphologie fonctionnelle	26
8 Descripteurs de texture	27
8.1 Statistiques	27
8.2 Motifs binaires locaux (LBP)	28
8.3 Méthodes fréquentielles	29
8.4 Méthodes spatiofréquentielles	29
9 Descripteurs de forme	30
9.1 Descripteurs classiques	30
9.2 Signature et descripteurs de Fourier	30
9.3 Codage de Freeman	30
9.4 Moments	31
9.5 Transformée de Hough	32
10 Compression d'images	33

11 Quelques références	34
-------------------------------	-----------

1 L'image : caractéristiques, représentations

Domaine spatial. L'image numérique est ici une matrice de *picture elements* ou pixels, chacun d'entre eux codant une nuance de couleur ou de niveau de gris. Cette image est acquise par un capteur relié à une carte d'acquisition disposant d'un convertisseur analogique numérique.

Une image est notamment caractérisée par ses résolutions spatiales et tonales. La première, appelée également *définition*, correspond tout simplement aux dimensions de l'image (hauteur ou nombre de lignes, largeur ou nombre de colonnes). La seconde résolution est liée au codage binaire de chaque pixel (en général 8 bits dans le cas d'une image en niveaux de gris et 24 dans le cas d'une image couleur).

Comme nous le verrons dans la section suivante, même si la plupart des images sont représentées dans l'espace couleur RGB, il existe différents modèles de représentation des couleurs. Quant au format d'enregistrement des images (dont .jpg, .png, .gif, etc), il s'agit d'un type de représentation de l'image, éventuellement avec compression, associée à des métadonnées (date, taille).

Une image peut-être vue comme un signal à deux dimensions. Alors qu'un signal évolue en fonction du temps (ou en fonction de la fréquence), l'image est une fonction de deux variables spatiales, qui peuvent être notées x et y , comme l'indique la figure 1 (gauche), et dont l'origine est placée en haut à gauche.

Domaine fréquentiel.

Comme tout signal, il existe une représentation fréquentielle de l'image, obtenue par transformée de Fourier, dont le calcul rapide s'appelle FFT pour *Fast Fourier Transform* (voir figure 1 (droite)). La FFT n'est possible que pour des images dont les dimensions sont des puissances de 2. Dans ce domaine fréquentiel, chaque point représente une fréquence particulière contenue dans le domaine spatial de l'image. Les axes u et v correspondent aux variables fréquentielles et l'origine $(0, 0)$ (composante continue) est placée au centre de cette représentation. L'axe horizontal u (resp. vertical v) est associé aux variations d'intensité lumineuse dans l'image le long des lignes (resp. des colonnes).

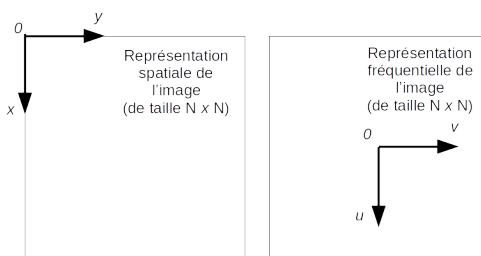


FIGURE 1 – Domaines spatial et fréquentiel de l'image.

La figure 2 d) montre un exemple de FFT calculée à partir de l'image 2 a). Les figures 2 b) et c) donnent un exemple de filtrage passe-bas. Le filtre utilisé est une gaussienne 2D centrée sur $(0, 0)$, qui par multiplication avec la FFT d'origine, supprime les variations d'intensité des fréquences situées au delà de la fréquence de coupure. Elle conserve les fréquences basses (les intensités moyennes). De façon similaire, les figures 2 e) et f) illustrent le filtrage passe-haut, qui permet d'isoler les fréquences hautes (voir figure 2 f)).

Histogramme. Parmi les représentations d'image, nous pouvons également ajouter l'histogramme $h(n)$, qui compte le nombre d'occurrences de chaque niveau de gris n d'une image (voir figure 3), et l'histogramme cumulé $hc(n)$ qui compte de nombre d'occurrences des pixels de niveau inférieur ou égal à n . Ces deux représentations s'avèrent très utiles pour analyser les statistiques de

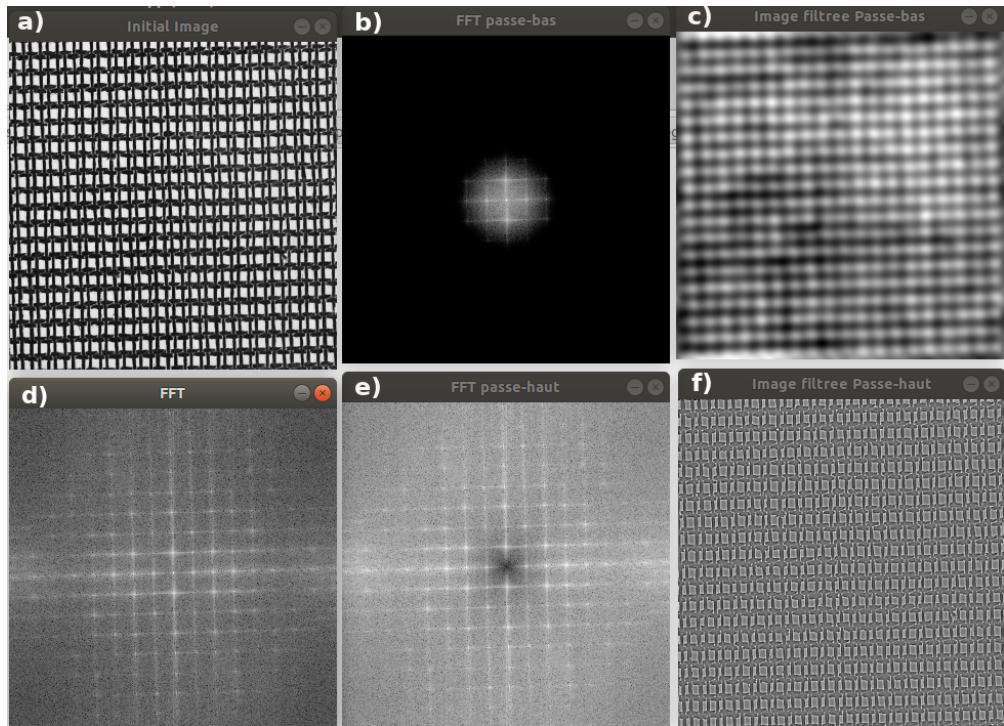


FIGURE 2 – La transformation de Fourier (FFT) d’une image. a) Image initiale dans sa représentation spatiale et d) son spectre d’amplitude (représentation fréquentielle), b) représentation fréquentielle de l’image avec filtrage passe-bas (les fréquences hautes sont atténuerées). c) image après filtrage passe-bas. e) filtrage passe-haut (les fréquences basses sont atténuerées). f) image après filtrage passe-haut.

l’images, améliorer le contraste de l’image (voir section 3), réaliser une classification ou une binarisation (voir dans la section 4). Les probabilités d’occurrence des niveaux de gris $p(n)$ se déduisent de l’histogramme en divisant chaque valeur par le nombre total de pixels de l’image (l’histogramme est alors normalisé).

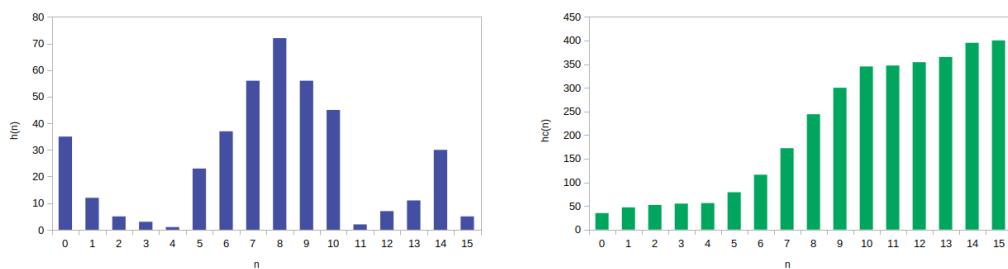


FIGURE 3 – Histogramme et histogramme cumulé d’une image de 400 pixels, avec une dynamique de 16 (on suppose que 4 bits sont utilisés pour le codage).

2 Couleur

La lumière est un ensemble d'ondes électromagnétiques, caractérisées par leurs longueurs d'ondes en nanomètres (nm). Le système visuel humain (SVH) est sensible aux ondes du spectre [380 -780 nm], le domaine du visible. Il s'agit alors de l'ensemble des couleurs monochromatiques sur la plage allant de 380 nm à 780nm (voir figure 5(a)).

2.1 Synthèse additive

Contrairement à ce que nous apprenons dès la maternelle sur le mélange des peintures primaires qui permettent de créer toutes les couleurs (par **synthèse soustractive**), les couleurs issues de rayons lumineux répondent à la **synthèse additive**. Trois couleurs, appelées primaires, sont suffisantes pour reproduire (par addition ou superposition) la quasi-totalité des couleurs visibles. Les 3 primaires sont choisies de manière à ce qu'aucune des 3 ne puisse être reproduite par un mélange des 2 autres. Un vecteur couleur est obtenu par une combinaison linéaire des 3 vecteurs de base :

$$\mathbf{C} = \alpha\mathbf{C}_1 + \beta\mathbf{C}_2 + \gamma\mathbf{C}_3$$

2.2 L'observateur standard de 1931

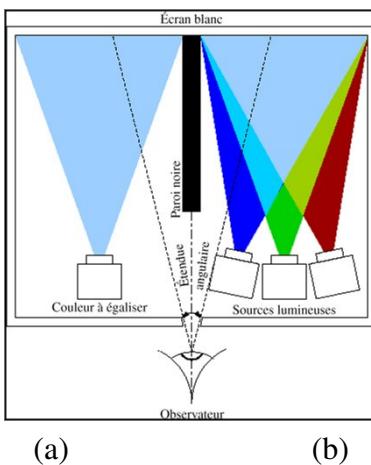


FIGURE 4 – Expérience d'égalisation.

Il est possible de déterminer expérimentalement la quantité des 3 primaires permettant une égalisation de chacun des rayonnements monochromatiques du visible (voir figure 4). Les expériences d'*égalisation* ont ainsi permis de définir les sensibilités colorées : les fonctions trichromatiques \bar{r} , \bar{g} , \bar{b} de l'observateur standard 1931. Ces expériences ont été menées par la Commission Internationale de l'Éclairage à partir d'un échantillon représentatif de la population.

L'égalisation est un procédé visant à reproduire visuellement, à partir d'un mélange pondéré de trois faisceaux de couleurs primaires, un stimulus chromatique de référence de longueur d'onde donnée. Plus concrètement, l'expérience était menée sur un échantillon représentatif de plusieurs personnes. On leur présentait deux signaux lumineux A et B. Le signal A est issu d'une lumière monochromatique et permettra de produire chaque couleur du spectre. Le signal B est produit en mélangeant (en additionnant) les flux lumineux produits par 3 spots, respectivement rouge, vert et bleu, et avec des intensités réglables. Ainsi, pour chaque couleur A, la personne doit trouver le mélange B aboutissant à la même couleur, en réglant l'intensité de chacun des 3 spots, intensités que l'on appelle \bar{r} , \bar{g} , \bar{b} .

Ainsi, les courbes de sensibilités $\bar{r}, \bar{g}, \bar{b}$ correspondent aux proportions de rouge, vert et bleu nécessaires pour reproduire l'ensemble des couleurs du visible (voir figure 5(a)).

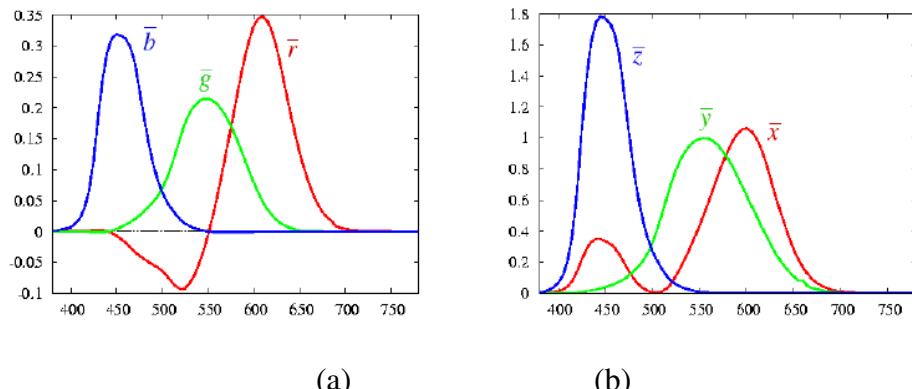


FIGURE 5 – Fonctions trichromatiques (a) $\bar{r}, \bar{g}, \bar{b}$ (b) $\bar{x}, \bar{y}, \bar{z}$.

2.3 La représentation des couleurs RGB

La figure 7 montre des exemples de types de capteurs produisant des composantes RGB qui seront manipulées en traitement d'images. Les composantes RGB sont représentées dans un espace cubique (voir figure 6).

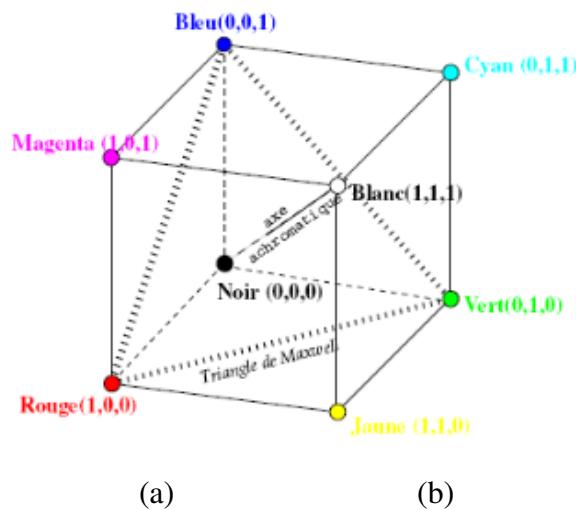


FIGURE 6 – Espace couleur de primaires RGB

Les composantes RGB proviennent de l'intégration, suivant les longueurs d'ondes du visible, du produit de 3 courbes :

1. les fonctions trichromatiques, c'est-à-dire l'observateur. Pour un capteur artificiel (caméra), les fonctions trichromatiques sont remplacées par les courbes de sensibilité spectrales de ce capteur (très variables d'un capteur à l'autre).
2. la réflectance du matériau \mathcal{R} (qui exprime pour chaque longueur d'onde, la proportion d'énergie réémise par la surface observée)
3. le spectre de l'illuminant \mathcal{E} (qui indique pour chaque longueur d'onde la quantité d'énergie émise).

On obtient alors les trois relations suivantes :

$$R = \int_{\lambda=380nm}^{780nm} \bar{r}(\lambda) \mathcal{R}(\lambda) \mathcal{E}(\lambda) d\lambda$$

$$G = \int_{\lambda=380nm}^{780nm} \bar{g}(\lambda) \mathcal{R}(\lambda) \mathcal{E}(\lambda) d\lambda$$

$$B = \int_{\lambda=380nm}^{780nm} \bar{b}(\lambda) \mathcal{R}(\lambda) \mathcal{E}(\lambda) d\lambda$$

Comme l'indique la figure 5(a), il n'est pas possible d'égaliser l'ensemble des couleurs monochromatiques à partir des composantes RGB. En effet, il n'est pas possible de produire les signaux lumineux rouges négatifs (entre 450 et 550nm). Pour pallier ce problème, la CIE a défini de nouvelles fonctions trichromatiques \bar{x} , \bar{y} , \bar{z} , par transformation linéaire des fonctions \bar{r} , \bar{g} , \bar{b} . Cela permet de définir un nouvel espace couleur de primaires : XYZ .

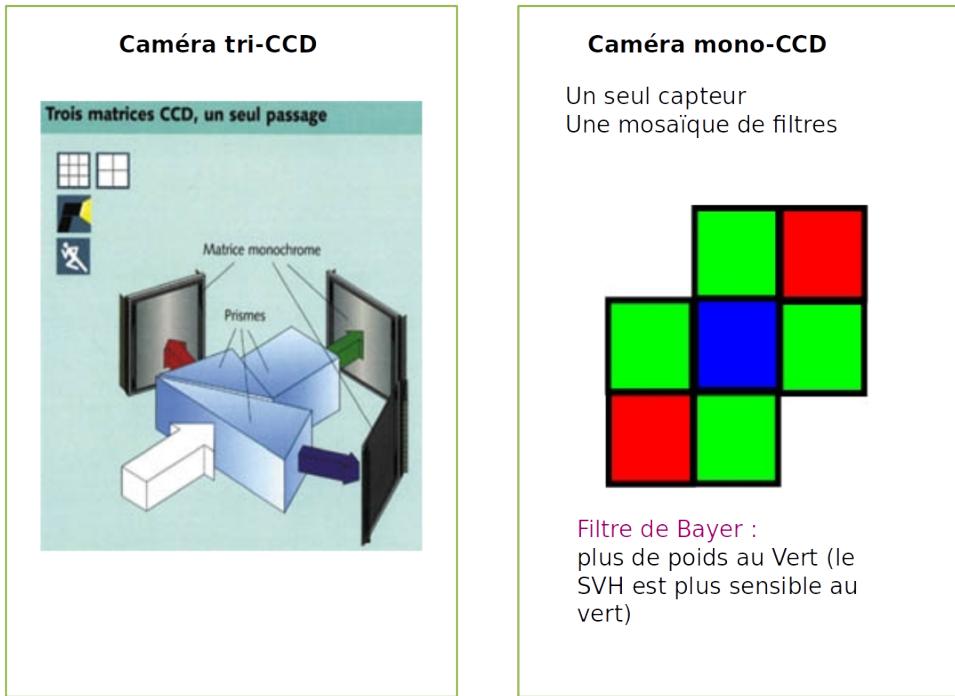


FIGURE 7 – Deux types de capteurs : tri-CCD et mono CCD.

2.4 Les espaces couleur

Par la suite, différents espaces de représentation de la couleur (voir quelques exemples sur la figure 8) ont été définis pour répondre à différentes problématiques :

- les **espaces perceptuels** (par exemple HSV, HSI) qui offrent une représentation cohérente avec la façon dont nous percevons et décrivons les couleurs, c'est-à-dire en trois composantes : la teinte (rouge, vert, jaune), la saturation (une couleur perçue comme *délavée* est faiblement saturée, une couleur dite *vive* est fortement saturée) et l'intensité ou luminance qui est liée à la quantité de lumière perçue. (voir figure 9 (a))
- les **espaces perceptuellement uniformes** (CIELAB ou $L^*a^*b^*$ et CIELUV) définis de façon à ce qu'une distance mesurée dans cet espace soit conforme à la distance perçue par le système visuel humain. Ces espaces sont particulièrement utiles pour comparer des couleurs entre elles. (voir figure 9 (b))

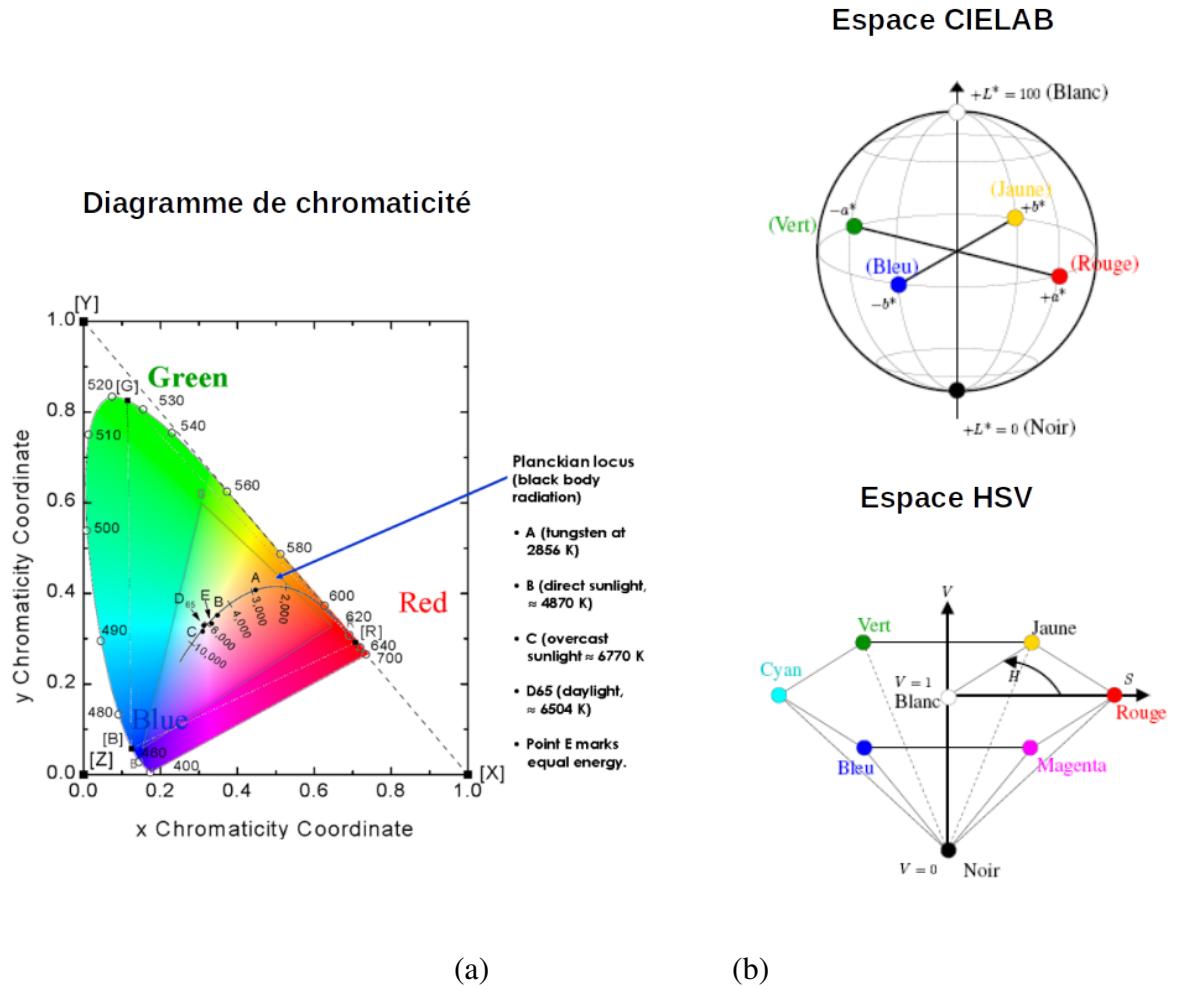


FIGURE 8 – Quelques espaces de représentation de la couleur.

- des **invariants couleur**, invariants par rapport aux changements d'éclairage et aux ombres, par exemple (voir figure 9 (c)) :

$$r = \frac{R}{R + G + B}, \quad g = \frac{G}{R + G + B} \text{ et } b = \frac{B}{R + G + B}$$

Notons que les espaces de type Teinte-Saturation-Luminance ainsi que CIELAB ou CIELUV montrent l'avantage de séparer les informations de **luminance**, susceptibles de varier avec les conditions d'éclairage, des informations de **chrominance** qui s'avèrent beaucoup plus stables vis-à-vis de ces perturbations. Contrairement aux espaces de primaires (RGB ou XYZ), les composantes sont bien moins corrélées.

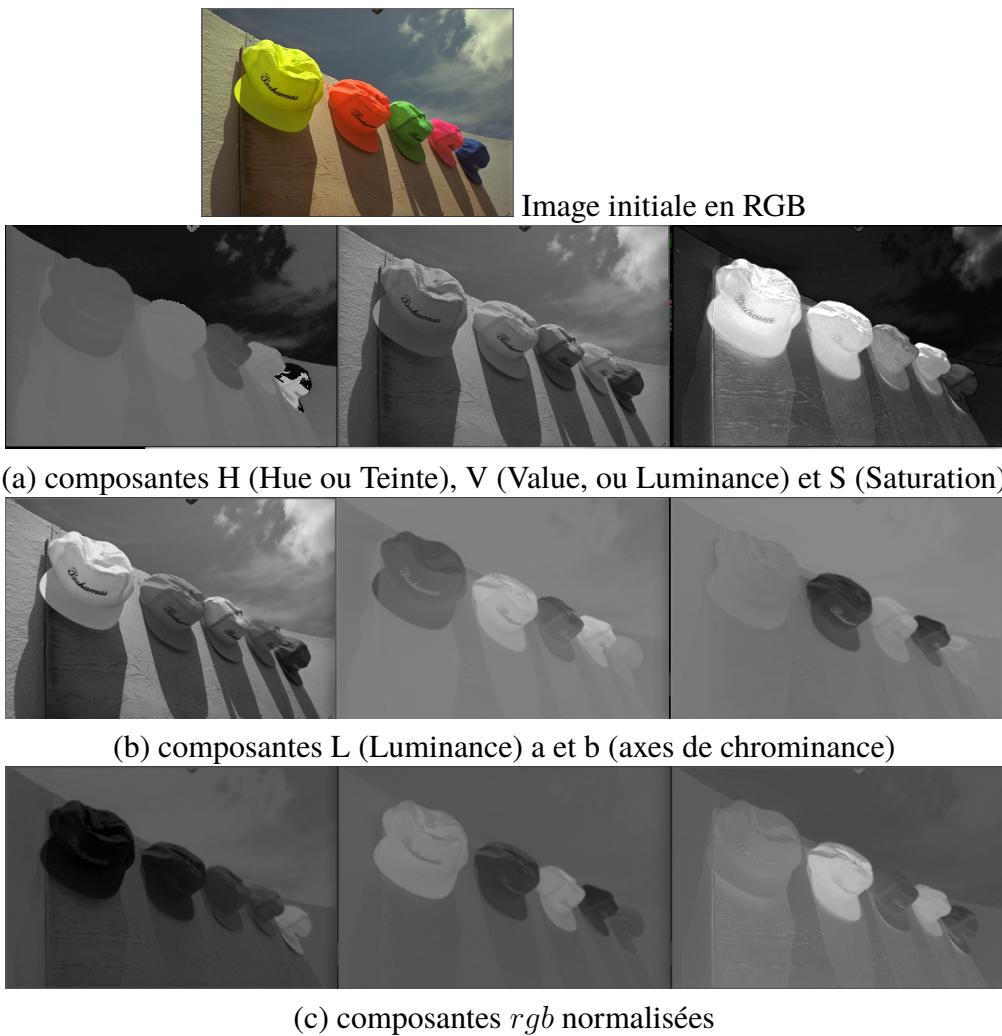


FIGURE 9 – Exemples de conversion d’images. (a) HSV, (b) CIELAB, (c) Norme L_1

3 Amélioration d’images

Nous verrons ici quelques méthodes simples de réhaussement de contraste et de réduction du bruit. Le contraste est la différence de luminosité entre les parties claires et sombres de l’image. Cette caractéristique se visualise bien sur l’histogramme. L’histogramme d’une image peu contrastée montre une distribution étroite, peu étalée, avec peu de variété d’intensités. Le bruit d’une image correspond à un défaut dégradant la qualité de l’image. Le bruit impulsionnel ou *poivre et sel* se caractérise par une altération de l’intensité de certains pixels, les faisant passer à 0 ou 255.

3.1 Réhaussement de contraste

Certaines modifications d’histogrammes permettent d’améliorer le contraste global d’une image ou encore d’améliorer le contraste pour certaines plages de niveaux de gris. On part d’une image $I(x, y)$ codée sur 8 bits dont on peut calculer un histogramme $h(n)$ où n correspond au niveau de gris allant de 0 à 255. On peut également calculer l’histogramme cumulé :

$$h_c(n) = \sum_{i=0}^n h(i) \quad \forall n \in [0, 255]$$

Attention, les transformations d’histogrammes ne s’appliquent que pour des images suffisamment riches en termes de contenu, comme des images *naturelles* (paysages par exemple). Dans le cas

contraire, c'est-à-dire pour des images présentant peu d'objets de luminance (ou de couleur) très uniforme, les modifications d'histogrammes risquent d'introduire du bruit en créant des variations artificielles de niveaux de gris.

3.1.1 Normalisation ou étirement d'histogramme

Elle consiste à exploiter toute la dynamique disponible (256 valeurs) par une simple normalisation qui ramène la valeur maximale de l'image \max à 255 et la valeur minimale \min à 0 :

$$J(x, y) = \frac{(I(x, y) - \min) \times 255}{\max - \min}$$

Toutefois, ce type de normalisation est très sensible au bruit. Par exemple, si l'image est très peu contrastée mais présente quelques pixels de bruit de type « poivre et sel » (c'est-à-dire noir et blanc), cette normalisation n'aura aucun effet. On préfère alors une normalisation qui exploite les statistiques globales de l'image : la moyenne d'intensité μ et l'écart-type σ (qui devra ensuite être suivie d'un nouvel étirement pour ramener les valeurs entre 0 et 255) :

$$J(x, y) = \frac{(I(x, y) - \mu)}{\sigma}$$

La figure 10 a) montre une image et son histogramme. Après étirement de l'histogramme (figure 10 b)), la dynamique est effectivement plus grande, et le contraste de l'image en est amélioré.

3.1.2 Étirement non-linéaire d'histogramme

De façon plus générale, l'étirement d'histogramme consiste à appliquer une fonction $J(x, y) = f(I(x, y))$ à chaque valeur de l'image $I(x, y)$ pour former une image *améliorée*. Dans le cas de la normalisation précédente, cette fonction était simplement une fonction linéaire (de la forme $ax + b$ c-a-d une droite). On peut choisir la forme de cette fonction pour réaliser une amélioration plus adaptée à un problème donné, ou à un type d'image donné. Si f est définie comme une fonction logarithmique (voir figure 10 c)), elle permet d'accentuer le contraste pour les valeurs de niveaux de gris faibles. Si elle est de forme exponentielle (voir figure 10 d)), elle permet d'accentuer le contraste pour les niveaux de gris élevés.

3.1.3 Égalisation d'histogramme

Comme son nom le laisse deviner, l'égalisation d'histogramme vise à obtenir un histogramme plat où toutes les valeurs de niveau de gris sont équiprobables $h(n) = N/256$ soit $p(n) = 1/256$ avec N le nombre de pixels de l'image. L'histogramme cumulé après égalisation forme une droite d'équation $N/256$ (ou $1/256$ en termes de probabilités). L'égalisation consiste à transformer l'histogramme cumulé vers cette équation de droite :

$$m = J(x, y) = f(n = I(x, y)) = \frac{255}{N} h_c(n)$$

où N est le nombre de pixels de l'image $I(x, y)$ et h_c est l'histogramme cumulé. Cette transformation permet de se ramener à une distribution équiprobable où chaque valeur de niveau de gris possède une probabilité de $1/256$. La figure 10 e) montre un exemple d'égalisation. La figure 11 montre les histogrammes de la figure 3 après égalisation.

3.2 Réduction du bruit, lissage

La réduction du bruit se fait par filtrage spatial, qui peut être linéaire (filtrage passe-bas) ou non-linéaire (filtrage d'ordre par exemple, comme le filtrage médian).

3 AMÉLIORATION D'IMAGES

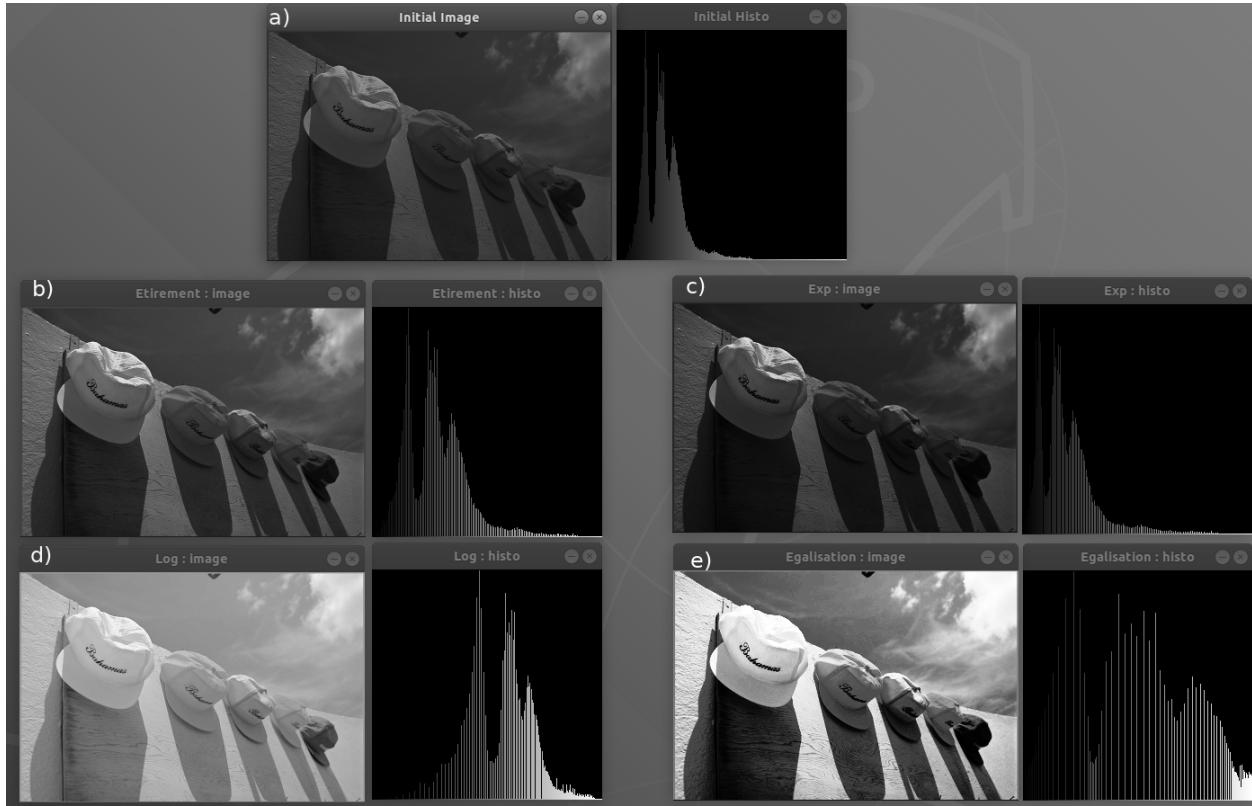


FIGURE 10 – Histogrammes et transformations. a) image et histogramme d'origine. b) étirement. c) transformation exponentielle. d) transformation log. e) égalisation.

3.2.1 Filtrage linéaire

Le **filtrage linéaire** est obtenu par convolution entre l'image initiale et un noyau (filtre) généralement **moyen, gaussien** (figure 13) ou **binomial**. La force (ou niveau) de filtrage dépend directement de la taille du noyau. Dans le cas du filtrage gaussien, la taille du noyau dépend directement du paramètre σ (noyau de largeur $> (2\sigma + 1)$). Le filtrage linéaire atténue l'amplitude du bruit (gaussien ou impulsif) mais lisse également les contours, puisqu'il atténue les fréquences hautes de l'image.

Gaussienne 1D centrée sur m :

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-m)^2}{2\sigma^2}}$$

Gaussienne 2D centrée sur (m_x, m_y) :

$$G(x, y) = G(x)G(y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-m_x)^2+(y-m_y)^2}{2\sigma^2}}$$

La figure 13 b) montre le résultat d'un filtrage passe-bas sur l'image de Lena bruitée avec du bruit impulsif (figure 13 a)). Le bruit impulsif est effectivement lissé, mais également les contours.

3.2.2 Filtrage d'ordre

Le **filtrage d'ordre** le plus connu et le plus utilisé est le **filtrage médian** dont la force dépend là aussi de la taille du filtre. Supposons un noyau de taille 3×3 centré sur le pixel à filtrer (ci-dessous le pixel de valeur 7). L'ensemble des 9 valeurs sont triées (peu importe l'ordre, qu'il soit croissant

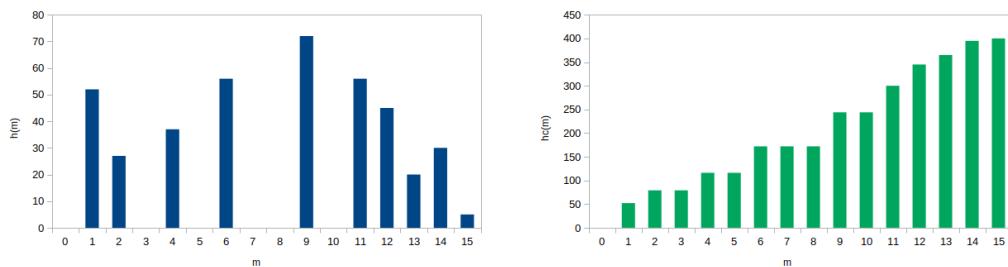


FIGURE 11 – Histogramme et histogramme cumulé d’une image de 400 pixels, après égalisation des histogrammes de la figure 3.

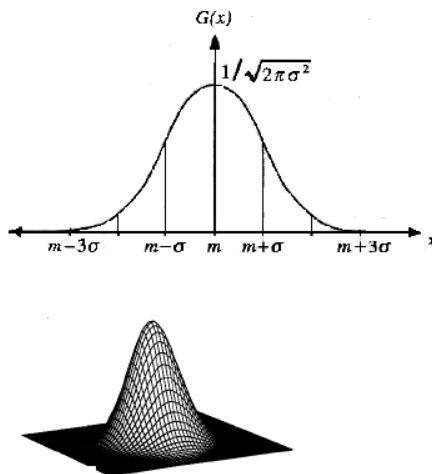


FIGURE 12 – Gaussiennes 1D et 2D.

ou décroissant) puis la valeur médiane est affectée au pixel courant dans l’image résultat :

$$\begin{array}{ccc}
 5 & 4 & 2 \\
 5 & 7 & 1 \rightarrow \text{tri} : 1, 2, 2, 4, 4, 5, 5, 5, 7 \rightarrow 5 & 4 & 1 \\
 5 & 4 & 2
 \end{array}
 \qquad
 \begin{array}{ccc}
 5 & 4 & 2 \\
 5 & 4 & 2
 \end{array}$$

Le filtrage médian est particulièrement adapté pour les images montrant du bruit impulsif ou poivre et sel. Il préserve bien les contours mais rogne les coins. La figure 13 c) illustre la bonne qualité du lissage médian sur l’image de Lena, dans le cas d’un bruit poivre et sel.



FIGURE 13 – Filtrage passe-bas. (a) image initiale avec bruit impulsionnel (poivre et sel). (b) résultat après filtrage gaussien. (c) Résultat après filtrage médian.

4 Classification

La classification consiste à partitionner les données en groupes (ou classes) d’attributs homogènes. Dans le cas d’une image, il s’agit généralement de classifier les couleurs ou les niveaux de gris, ou encore des attributs tels que le mouvement, des attributs de texture, etc. La plupart des méthodes de classification travaillent uniquement à partir des attributs des pixels sans considérer leur distribution spatiale dans l’image ou encore leur connexité. Dans ce sens, les méthodes de classification se distinguent des méthodes de segmentation qui partitionnent l’image en régions disjointes de pixels connexes.

4.1 Binarisation

La binarisation peut être vue comme une classification en 2 classes. La plus simple des techniques consiste à définir des seuils fixes. Toutes les valeurs inférieures, supérieures ou situées entre deux seuils sont mises à 1 et le reste à 0. Comme il est souvent difficile de définir la ou les valeurs de seuils, des méthodes de seuillage automatique sont préférables :

- Le seuillage au **percentile** supérieur ou inférieur. Un percentile correspond à chacune des 99 valeurs qui divisent les données triées en 100 parts égales. Le seuillage supérieur au p-ième percentile consiste à mettre à 1 les pixels dont la valeur dépasse celle des p% de pixels les plus faibles.
- Seuillage automatique de **Otsu**. Le seuil est calculé de façon à minimiser les variance intra-classe et maximiser la variance inter-classe.

Le seuillage par **hystérosis**, décrit par l’algorithme 1 sort un peu du cadre de la classification dans le sens où il exploite la connexité. On parle de seuillage local. Ce seuillage utilise deux seuils, un seuil haut S_h et un seuil bas S_b . Les valeurs de pixels $> S_h$ sont mises à 1. Les valeurs $< S_b$ sont mises à 0. Enfin, les valeurs intermédiaires sont mises à 1 lorsqu’elles forment une composante connexe qui est elle-même connexe avec une composante $> S_h$. Ce type de seuillage est notamment utilisé pour la sélection des contours à partir d’un calcul de gradient (voir la partie 6.1).

4.2 Clustering

K-means. La méthode des nuées dynamiques (k-means) proposée par Forgy¹ est l’approche la

1. R. Forgy, Cluster Analysis of Multivariate Data : Efficiency versus Inter-pretability of Classification, Biometrics

Algorithm 1 Exemple d'algorithme de seuillage par hystérésis

Entrée : image de niveaux de gris I , seuil haut S_h , seuil bas S_b

Sortie : image binaire B

$I_h \leftarrow \{1 \text{ si } I > S_h, 0 \text{ sinon}\}$

$I_b \leftarrow \{1 \text{ si } I > S_b, 0 \text{ sinon}\}$

$n_{chts} \leftarrow 1$

while $n_{chts} > 0$ **do**

$I_{temp} \leftarrow \text{dilate}(I_h)$

$I_{temp} \leftarrow I_b \text{ AND } I_{temp}$

$n_{chts} \leftarrow \text{comptage_pixels_non_nuls}(I_h - I_{temp})$

$I_h \leftarrow I_{temp}$

end while

plus courante (voir l'algorithme 2). Le critère d'arrêt est fondé sur la stabilisation des classes, qui peut être détectée en scrutant le nombre de changements d'affectation de classe entre deux itérations ou en calculant la variance interclasses. L'algorithme s'arrête par exemple lorsque la variance interclasses n'évolue plus significativement. La figure 14 donne un exemple de classification sur une image de panneaux routiers, préalablement convertie dans l'espace rgb (composantes normalisées) pour une meilleure qualité de résultat.

Algorithm 2 Nuées dynamiques (k-means)

Initialisation : Tirer au hasard, ou sélectionner pour des raisons extérieures à la méthode, k points dans l'espace des individus, en général k individus de l'ensemble, appelés centres ou noyaux.

while les classes ne sont pas stabilisées **do**

- Allouer chaque individu au centre (c'est-à-dire à la classe) le plus proche au sens de la métrique euclidienne choisie ; on obtient ainsi, à chaque étape, une classification en k classes, ou moins si, finalement, une des classes devient vide.

- Calculer le centre de gravité de chaque classe : il devient le nouveau noyau ; si une classe s'est vidée, on peut éventuellement retirer aléatoirement un noyau complémentaire.

end while



FIGURE 14 – Exemple de classification couleur en 3 classes.

ISODATA. La méthode k-means nécessite de préciser le nombre de classes prévues. L'algorithme **ISODATA** (pour Iterative Self-Organizing Data ANalysis Technique) proposé par Jensen en 1996 permet de pallier ce problème (mais des paramètres sont toutefois requis). <https://www.cs.umd.edu/~mount/Projects/ISODATA/ijcga07-isodata.pdf>

Mean-Shift. L'algorithme de clustering **mean-shift** consiste à rechercher les modes dans une distribution, c'est-à-dire les maxima locaux. Cet algorithme est bien illustré sur cette page <https://spin.atomicobject.com/2015/05/26/mean-shift-clustering/>. Comme ISODATA, le nombre de classes n'est pas requis.

5 Régions

5.1 Étiquetage en composantes connexes

Il existe de très nombreuses variantes d’algorithmes d’étiquetage en composantes connexes. La thèse de Laurent Cabaret² en fait un état de l’art détaillé. L’algorithme 3 donne une version qui parcourt l’image binaire dans le sens du balayage vidéo classique. Dès qu’un pixel non nul est trouvé, la composante connexe est reconstruite de proche en proche en agrégeant à la région l’ensemble des pixels connexes et non-nuls. Une pile est utilisée pour stocker l’ensemble des pixels à agréger à la région, afin que leurs voisinages soient analysés à leur tour (algorithme 3).

La figure 15 montre un exemple d’étiquetage en composantes connexes sur une image de panneaux préalablement binarisée.

Algorithm 3 Étiquetage en composantes connexes $E, n = ECC(I, e)$

Entrée : image binaire I , étiquette initiale e
Sortie : image d’étiquettes E , nombre de régions n
Initialisation
 $ToDo \leftarrow I$
 $E(x, y) \leftarrow 0 \forall (x, y)$
Liste de pixels $L \leftarrow 0$
for $x = 0$ to rows **do**
 for $y = 0$ to cols **do**
 if $I(x, y) \neq 0$ **and** $ToDo(x, y) \neq 0$ **then**
 $L \leftarrow (x, y)$
 $e = e + 1$
 $E(x, y) = e$ //nouvelle composante connexe
 $ToDo(x, y) = 0$
 repeat
 $(u, v) \leftarrow \text{pop}(L)$
 for $i = -1$ to 1 **do**
 for $j = -1$ to 1 **do**
 if $(i \neq 0$ **or** $j \neq 0)$ **and** $(I(i + u, j + v) \neq 0$ **and** $ToDo(i + u, j + v) \neq 0)$ **then**
 $L \leftarrow (i + u, j + v)$
 $E(i + u, j + v) = e$
 $ToDo(i + u, j + v) = 0$
 end if
 end for
 end for
 until L est non vide
 $n \leftarrow e$
 end if
 end for
end for

5.2 Segmentation après une classification

La segmentation peut être effectuée après une première étape de classification. Par exemple, après l’algorithme k-means, k classes de pixels sont extraites. Pour chacune d’entre elles, on met à

2. <https://tel.archives-ouvertes.fr/tel-01597903>



FIGURE 15 – Exemple de segmentation par étiquetage en composantes connexes.

1 l'ensemble des pixels associés à la classe puis l'on opère un étiquetage (algorithme 4).

Algorithm 4 Segmentation après classification

Entrée : image de K classes C

Sortie : image d'étiquettes E , nombre de régions n

Initialisation

$E(x, y) \leftarrow 0 \forall (x, y)$

$n = 0$

for $k = 1$ **to** K **do**

$I(x, y) \leftarrow 1$ si $C(x, y) == k \forall (x, y)$

$E, n \leftarrow ECC(I, n)$

end for

5.3 Croissance de régions

L'algorithme de croissance de régions est assez similaire à celui de l'étiquetage expliqué précédemment (algorithme 3). Ici nous ne raisonnons pas à partir d'une image binaire, mais d'une image en niveaux de gris ou en couleur. La différence se porte sur le test à réaliser pour agréger un nouveau pixel à la région en cours de formation, le test souligné dans l'algorithme 3, qui devient : **if** ($i \neq 0$ ou $j \neq 0$) **and** (similarity($I(i + u, j + v), I(u, v)$) **and** $ToDo(i + u, j + v) \neq 0$)

La fonction *similarity* renvoie 1 si les deux valeurs image sont suffisamment proches, par exemple si la distance euclidienne entre les 2 couleurs est inférieure à un seuil.

Il existe un grand nombre d'approches de segmentation d'image, comme les approches **ligne de partage des eaux** (ou watershed)³.

L'image est considérée comme un relief topographique, avec des bassins versants séparés par une ligne de partage des eaux. Ce type d'algorithme aboutit généralement à une sur-segmentation, c'est-à-dire que chaque objet réellement présent dans l'image est constitué de plusieurs régions dans l'image. Ces régions peuvent ensuite être fusionnées entre elles pour constituer des régions plus cohérentes avec le contenu image.

3. voir par exemple la thèse <https://tel.archives-ouvertes.fr/tel-00321885/document> et la documentation d'opencv à ce sujet : https://docs.opencv.org/3.4/d3/db4/tutorial_py_watershed.html

6 Contours

Les contours de l'image correspondent à des pixels de fort gradient d'intensité. Déetecter les contours peut donc être faire en calculant le gradient image et en sélectionnant les valeurs de fort gradient par seuillage (seuillage simple comme sur la figure 16 b) ou seuillage par hystérésis (figure 16 (c)) décrit à la section 4.1) ou en détectant les maxima locaux (voir paragraphe 6.1).

Les contours peuvent être extraits à partir des dérivées secondes de l'image (le Laplacien). Dans ce cas, on détecte les passages par zéro du laplacien (voir paragraphe 6.2 et figure 16 (d)).

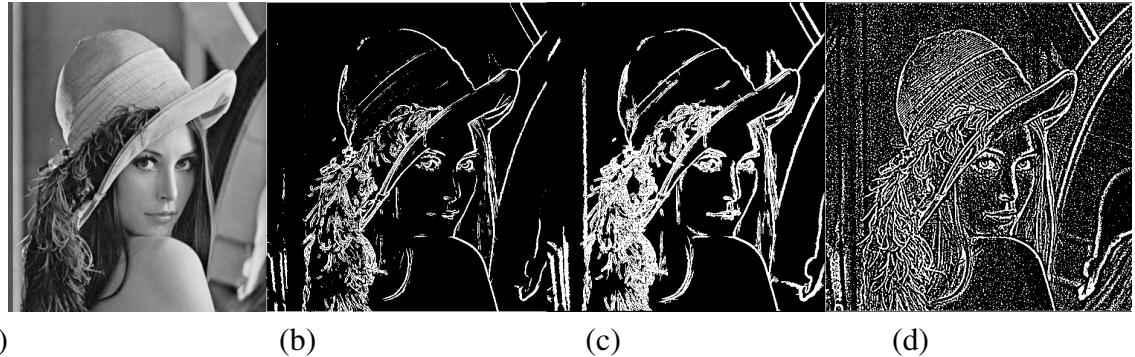


FIGURE 16 – Détection de contours. (a) Image initiale. (b) Contours par seuillage simple de la norme du gradient. (c) seuillage par hystérésis. (d) Détection des contours par détection des passages par zéro du Laplacien.

6.1 À partir du gradient image

Le gradient est un vecteur à deux composantes, soit (G_x, G_y) où G_x et G_y sont les deux dérivées de l'image suivant les deux directions x et y , soit (n, θ) où n est la norme (le module du vecteur) et θ l'orientation :

$$n = |G| = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \text{atan} \left(\frac{G_y}{G_x} \right)$$

Le calcul de gradient le plus simple se calcule par simple différence :

$$G_x = I(x+1, y) - I(x, y) \text{ et } G_y = I(x, y+1) - I(x, y)$$

Ce calcul s'avère assez sensible au bruit. On préfère donc les détecteurs par convolution (filtrage linéaire).

Détecteur par filtrage linéaire. L'image est convoluée (opérateur $*$) par un noyau ou filtre passe-haut par direction $G_x = I * K_x$ et $G_y = I * K_y$. La plupart d'entre eux combinent un filtrage passe-bas avec un filtrage passe-haut de façon à atténuer les gradients liés au bruit. Le filtre le plus courant est celui de Sobel :

$$K_x = \frac{1}{4} \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad K_y = \frac{1}{4} \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Le détecteur de Prewitt fait appel à 4 filtres (pour 4 directions). La norme du gradient est finalement calculée comme le maximum des 4 réponses des convolutions avec les 4 noyaux suivants :

$$K_1 = \frac{1}{3} \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}, K_2 = \frac{1}{3} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{pmatrix}, K_3 = \frac{1}{3} \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -1 \end{pmatrix}, K_4 = \frac{1}{3} \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix}$$

Détecteur de Canny⁴ Il se fonde sur quatre étapes :

1. Filtrage Gaussien de l'image avec un paramètre σ
2. Calcul du gradient ($|G|, \theta$) en utilisant les noyaux $K_x = [-1, 0, 1]$ et $K_y = [-1, 0, 1]^\top$
3. Détection des maxima locaux
4. Seuillage par hystérosis, qui nécessite deux seuils (voir section 4)

6.2 À partir du Laplacien

6.2.1 Laplacien

Le laplacien (calculé à partir des dérivées secondes) de l'image est défini comme suit :

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

Ce laplacien est nul lorsque les deux dérivées secondes (en x et y) sont nulles. L'opérateur est linéaire et symétrique en rotation. Une façon dite *naïve* de le calculer est la suivante :

$$\begin{aligned} I''_x &\simeq I'_x - I'_{x-1} \\ &\simeq (I(x+1) - I(x)) - (I(x) - I(x-1)) \\ &\simeq I(x+1) - 2I(x) + I(x-1) \end{aligned}$$

Le filtre Laplacien peut être défini en 4 ou 8 connexité :

$$\text{4 connexité : } \frac{1}{4} \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix} \quad \text{8 connexité : } \frac{1}{8} \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix}$$

6.2.2 Laplacien de la Gaussiennce : LOG et DOG

La convolution étant **associative** : $(I * g) * f = I * (g * f)$, il est possible de réaliser un lissage et une dérivation à l'aide d'un même filtre au lieu de réaliser 2 filtrages successifs. Ce noyau de lissage-laplacien est obtenu en calculant la dérivée seconde du filtre de lissage : le laplacien du filtre gaussien (voir figure 17)

Le laplacien de la gaussienne centré sur 0 et d'écart-type σ s'exprime par :

$$LoG(x, y) = -\frac{4}{\sqrt{2\pi\sigma^2}} \left(\frac{(x - m_x)^2 + (y - m_y)^2}{\sigma^2} - 1 \right) \exp -\frac{(x - m_x)^2 + (y - m_y)^2}{2\sigma^2}$$

Il s'agit d'un opérateur anisotropique pour lequel on ne peut pas calculer la direction du contour (contrairement au gradient).

4. Canny, J., A Computational Approach To Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, 8 :679-698 (1986).

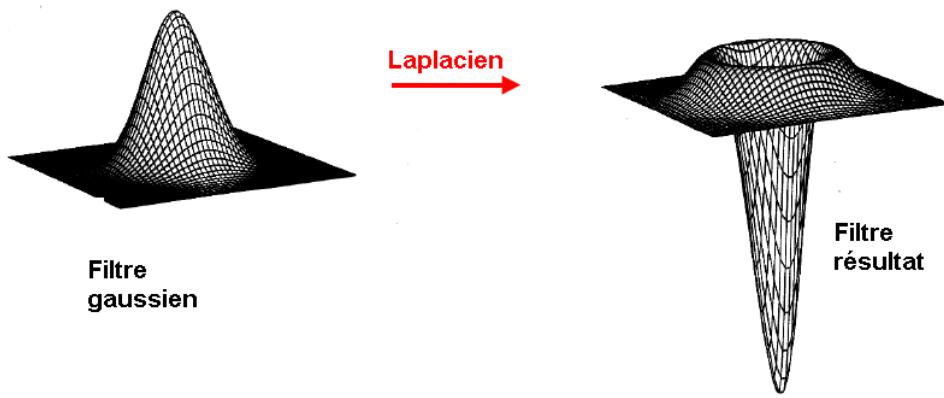


FIGURE 17 – Gaussienne et Laplacien de la gaussienne.

L'opérateur DoG (*Difference of Gaussians*) est une variante plus rapide de l'opérateur *LoG*. Il approxime le filtre *LoG* par la différence de deux gaussiennes de tailles différentes.

À partir de la réponse au filtre *LoG* ou *DoG*, les contours sont localisés en détectant les passages par zéro. En effet, les maxima locaux du gradient se caractérisent par un passage par zéro de la dérivée seconde. Cela peut se faire par exemple à l'aide de l'algorithme 5. Une autre méthode, qui exploite la morphologie mathématique (voir section 7) consiste à détecter les ensembles de pixels de valeurs < 0 puis de détecter le contours de ces ensembles (par un gradient morphologique).

Les zéros constituent un réseau de lignes fermées, ce qui permet d'éviter des post-traitements de fermeture de contours, pour réaliser par exemple une analyse des formes.

Algorithm 5 Localisation des contours à partir des dérivées secondes : passage par 0

```

Entrée : image de Laplacien  $L$ 
Sortie : image binaire  $C$  de contours
 $C \leftarrow 0$ 
for  $x = 0$  to rows -1 do
    for  $y = 0$  to cols -1 do
        if  $(L(x, y) \times L(x+1, y) > S \text{ or } L(x, y) \times L(x+1, y+1) > S \text{ or } L(x, y) \times L(x, y+1) > S)$ 
        then
             $C(x, y) = 1$ 
        end if
    end for
end for
```

6.3 Lissage et dérivation

Si l'opérateur de dérivation est linéaire \Rightarrow opérateur associatif et commutatif (convolution). Soit $I(x, y)$ (l'image) et $f(x, y)$ (un filtre de lissage) deux fonctions L^2 dérivables avec l'opérateur $\mathcal{O}_{x,y}$ (dérivation) alors :

$$\mathcal{O}_{x,y}(I(x, y) * f(x, y)) = \mathcal{O}_{x,y}(I(x, y)) * (x, y) = I(x, y) * \mathcal{O}_{x,y}(f(x, y))$$

Dans le cas d'un **filtrage linéaire** : l'ordre dans lequel se font le lissage et la dérivation n'a pas d'importance (convolution). Dans le cas d'un filtrage non-linéaire il est préférable de faire le lissage avant la détection des gradients.

6.4 Autres détecteurs

Gradients morphologiques (voir la section 7) définis à l'aide de l'érosion et de la dilatation.

7 Morphologie mathématique

La théorie de la morphologie mathématique s'appuie sur la théorie des ensembles. Elle regroupe de nombreuses transformations non-linéaires à partir d'un objet de référence (élément structurant B) permettant d'extraire différentes informations telles que la connexité, la taille, la forme, l'orientation. Il s'agit d'étudier ou de traiter un ensemble (les objets dans les images) à l'aide d'un autre ensemble, appelé élément structurant, qui sert de sonde. L'élément structurant est déplacé en tout point de l'image à traiter. À chaque position on étudie sa relation avec l'image binaire X : « est inclus dans l'ensemble », « intersecte l'ensemble », par exemple. Les transformations se déduisent par composition ou différences d'opérations élémentaires : l'érosion et la dilatation.

Élément structurant. L'élément structurant peut avoir n'importe quelle forme et taille. Néanmoins les plus utilisés sont la croix (pour l'analyse en 4-connexité) et le carré (8-connexité), mais ce peut être une ligne, une colonne etc suivant l'objectif visé. L'élément structurant possède un centre, qui est le point de référence pour l'analyse de la relation avec l'image binaire. C'est lui qui est positionné sur chaque pixel de l'image à traiter.

$$B_4 = \begin{array}{|c|c|c|} \hline & 1 & \\ \hline 1 & \textbf{1} & 1 \\ \hline & 1 & \\ \hline \end{array} \quad B_8 = \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & \textbf{1} & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Considérons l'espace image à deux dimensions $E = \mathbb{Z}^2$ et B est un sous-ensemble de E . On appelle X un ensemble dans l'image à analyser $\in E$.

Le translaté de B à la position x s'écrit $B_x = \{b + x | b \in B\}$.

7.1 Dilatation et érosion

La **dilatation** permet d'étudier les relations « intersecte avec l'ensemble ». La dilatation morphologique avec l'élément structurant B est définie comme la somme de Minkowski :

$$\delta_B(X) = X \oplus B = \{x + b | b \in B, x \in X\} = \cup_{x \in X} B_x$$

De manière plus intuitive : le centre de B est déplacé en chaque point de l'image. S'il y a intersection entre un élément b de B et un élément x de l'ensemble X , le résultat est égal à 1 et 0 sinon.

L'**érosion** permet d'étudier les relations « est inclus dans l'ensemble »

$$\epsilon_B(X) = X \ominus B = \{x | B_x \subset X\}$$

De manière plus intuitive : le centre de B est déplacé en chaque point de l'image. Si B est totalement inclus dans X alors le résultat est 1, sinon 0.

0 0 0 0 0 0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0 0 0 0 0 0	
0 0 1 1 1 0 0 0 0 0 0 0 0	
0 0 1 0 1 0 0 0 0 0 0 0 0	
0 0 1 1 1 1 0 0 0 0 0 0 0	
0 0 1 1 1 0 0 0 0 0 0 0 0	
0 0 1 1 1 0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0 0 0 0 0 0	
0 0 0 0 0 0 0 0 0 0 0 0 0	

Image binaire initiale I

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Érosion de I avec B_8

0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0	0	0	0
0	1	1	1	1	1	1	0	0	0	1	1	1
0	1	1	1	1	1	1	1	0	1	1	1	0
0	1	1	1	1	1	1	1	0	1	1	1	0
0	1	1	1	1	1	1	1	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Dilatation de I avec B_8

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Érosion de I avec B_4

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	0	0
0	1	1	1	1	1	1	1	0	0	0	1	0
0	1	1	1	1	1	1	1	0	0	1	1	1
0	1	1	1	1	1	1	1	1	0	0	1	0
0	1	1	1	1	1	1	1	1	0	0	1	0
0	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Dilatation de I avec B_4

7.2 Ouverture et fermeture

Ouverture. Érosion puis dilatation. $\gamma_B(X) = X \circ B = \delta_B(\epsilon_B(X))$. L'ouverture élimine les ensembles de taille inférieure à celle de l'élément structurant.

Fermeture. Dilatation puis Érosion. $\phi_B(X) = X \bullet B = \epsilon_B(\delta_B(X))$. La fermeture élimine les trous de taille inférieure à celle de l'élément structurant.

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0
0	0	1	1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Ouverture de I avec B_8

0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	1	1	1	1	1	1	0	0	0	0	0
0	0	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0

Fermeture de I avec B_8

7.3 Gradients morphologiques

- Gradient morphologique : $\delta_B(X) - \epsilon_B(X)$
- Contours internes : $X - \epsilon_B(X)$ (Image - image érodée).
- Contours externes : $\delta_B(X) - X$ (Image dilatée - image)

7.4 Transformation en tout ou rien (*Hit or Miss*)

La transformation en tout ou rien sert à détecter des configurations particulières de pixels, par exemple des coins. Elle utilise 2 éléments structurants notés ici A et B . On vérifie que A est exclus

de X et que B est inclus dans X . Remarquons que vérifier que A est exclus de X revient à vérifier que A est inclus dans le complémentaire de X , noté X^c .

$$\eta(X) = \{x \mid A_x \subset X^c; B_x \subset X\}$$

Par exemple, pour détecter des coins sortants situés en haut à gauche des objets, on peut utiliser les deux éléments structurants suivants :

$$A = \begin{array}{ccc} 1 & 1 & 1 \\ 1 & \square & \\ 1 & & \end{array} \quad B = \begin{array}{cc} [1] & 1 \\ 1 & \end{array}$$

On remarquera qu'ici, le centre de l'élément structurant A est situé à l'extérieur.

7.5 Squelette morphologique

Le squelette morphologique (euclidien) est la réunion des centres de boules (euclidiennes) maximales, comme l'illustre la figure 18. On définit un élément structurant de type boule (le carré en est une approximation) de rayon R (carré de taille $2R+1 \times 2R+1$), que nous notons B^R . Considérons également un élément structurant boule de rayon 1, que nous notons B^1 .

Pour des valeurs de R croissantes :

- On effectue l'érosion de X par B^R : $E = \epsilon_{B^R}(X)$
- Par ailleurs, on applique l'ouverture de E par B^1 : $O = \gamma_{B^1}(E)$
- Le squelette est défini comme l'ensemble des points appartenant à E et non à O :

$$S = \{x \subset X \mid x \subset E \text{ and } x \not\subset O\}$$

La valeur de R est augmentée progressivement. À chaque itération, les points qui satisfont les critères (appartenance à E et non appartenance à O) sont conservés. L'algorithme s'arrête lorsque la largeur de la boule ($2R+1$) devient supérieure à l'épaisseur maximale de l'objet.

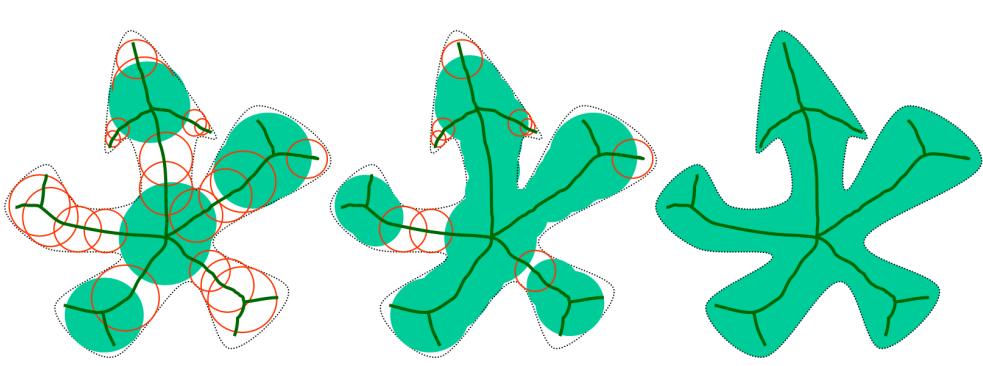


FIGURE 18 – Principe de la construction d'un squelette morphologique à partir de la boule de rayon maximal.

Référence : <https://perso.telecom-paristech.fr/bloch/P6Image/Squelettes.pdf>

7.6 Reconstruction géodésique et seuillage par hystérésis

On considère d'une part une image binaire I_1 contenant plusieurs ensembles connexes de laquelle nous voulons en sélectionner un sous-ensemble. Une seconde, I_2 contient des *marqueurs*

(pixels à 1) positionnés sur les ensembles à sélectionner dans I_1 . La reconstruction géodésique consiste à reconstruire les ensembles de I_1 marqués par les pixels de I_2 . L'algorithme 6 en propose une implémentation. Il s'agit de partir de l'image des marqueurs I_2 , de la dilater jusqu'à couvrir l'ensemble des régions à sélectionner dans I_1 .

Le seuillage par hystérésis, évoqué dans la section 4, est un exemple de reconstruction géodésique, où les marqueurs I_2 sont les pixels supérieurs à un seuil haut et I_1 les pixels supérieurs à un seuil bas.

Algorithm 6 Reconstruction géodésique binaire

Entrée : image binaire I_1 , et image de marqueurs I_2

Sortie : image binaire R

$R \leftarrow I_2$

repeat

 tmp \leftarrow Dilate (R) //dilatation avec élément carré 3×3

 tmp \leftarrow tmp AND I_1

$n_{chg} \leftarrow$ sum(tmp XOR R) //compte le nombre de changements

$R \leftarrow$ tmp

until $n_{chg} > 0$

7.7 Morphologie fonctionnelle

Une image à niveaux de gris peut être modélisée comme une fonction de \mathbb{Z}^2 dans \mathbb{Z} . Soit f une fonction appartenant à cet ensemble.

L'érosion se définit comme $\epsilon_B(f) = \inf\{f_b \mid b \in B\}$. L'élément structurant est déplacé en chaque point de l'image. En chaque position, la valeur minimale de l'image dans l'ensemble défini par B est recherché et affecté à l'image résultat. De façon similaire, la dilatation consiste à affecter la valeur maximale : $\delta_B(f) = \sup\{f_b \mid b \in B\}$

8 Descripteurs de texture

8.1 Statistiques

8.1.1 Statistiques du 1er ordre

Ces statistiques se déduisent de la probabilité $p(n)$ du niveau de gris n ou de l'histogramme $h(n) \simeq Np(n)$, avec N le nombre de pixels de l'image.

- **Moments d'ordre k**

$$\mu_k = \sum_n n^k p(n)$$

En particulier μ_1 correspond à la moyenne des niveaux de gris

- **Moments centrés d'ordre k**

$$\eta_k = \sum_n (n - \mu_1)^k p(n)$$

Variance $\sigma^2 = \eta_2$: écart des variations d'intensités par rapport à la moyenne.

Skewness $\gamma_1 = \eta_3$, nul lorsque l'histogramme est symétrique par rapport à la moyenne

Kurtosis (aplatissement) $\gamma_2 = \eta_4$

- **Énergie**

$$W = \sum_n |p(n)|^2$$

L'énergie est élevée pour une image homogène et minimale lorsque les probabilités des niveaux de gris sont équiréparties.

- **Entropie**

$$E = - \sum_n p(n) \log p(n)$$

L'entropie est faible pour une image homogène et élevée pour un histogramme étiré.

- **Contraste**

$$C = (\max(n) - \min(n)) / (\max(n) + \min(n))$$

Le contraste est faible pour une image uniforme.

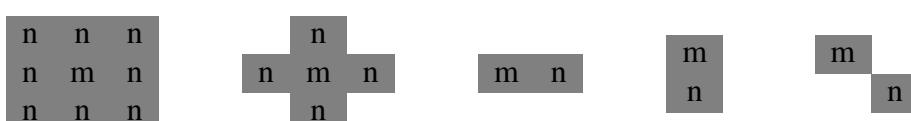
- **Dynamique**

$$D = \max(n) - \min(n)$$

La dynamique est faible pour une image uniforme.

8.1.2 Matrice de cooccurrence et statistiques du 2^e ordre

L'algorithme 7 décrit le calcul de la matrice de cooccurrence. On considère un voisinage V qui peut être de forme variée (à choisir en fonction de la tâche visée) :



Comme pour les statistiques extraites à partir de l'histogramme, il est possible de décrire les matrices de cooccurrence à l'aide de statistiques, où $p(m, n) = M(m, n)/N$ où N est égal au nombre de pixels de l'image \times le nombre de voisins dans V .

Algorithm 7 Matrice de cooccurrence $M(m,n)$

Input : image I
Output : matrice de cooccurrence M
for $(x, y) \in I$ **do**
 for $(u, v) \in V(x, y)$ **do**
 $m \leftarrow I(x, y)$
 $n \leftarrow I(u, v)$
 $M(m, n) \leftarrow M(m, n) + 1$
 end for
end for

Moyennes $\mu_m = \sum mp(m, n)$ et $\mu_n = \sum np(m, n)$

Moments centrés $\eta_{kl} = \sum_m \sum_n (m - \mu_m)^k (n - \mu_n)^l p(m, n)$

Énergie $W = \sum \sum (p(m, n))^2$

Entropie $E = - \sum \sum p(m, n) \log(p(m, n))$

Contraste $C = \sum_m \sum_n (n - m)^2 p(m, n)$

Homogénéité $H = \sum_m \sum_n \frac{1}{1 + (m - n)^2} p(m, n)$

8.2 Motifs binaires locaux (LBP)

Les LBP codent les différences locales entre niveaux de gris. On considère un voisinage de forme circulaire de rayon R (un cercle constitué de P points), autour du point central de coordonnées (x_c, y_c) et de niveau de gris g_c

$$LBP(x_c, y_c) = \sum_{p=0}^{P-1} 2^p \delta(g_p - g_c)$$

où δ est la fonction de Heavyside, à savoir que $\delta(g_p - g_c) = 1$ si $g_p \geq g_c$ et 0 sinon. Pour un voisinage de 8 points sur un rayon $R = 1$, les valeurs de LBP sont codées entre 0 et 255. Une texture peut être caractérisée par l'histogramme de ces LBP.

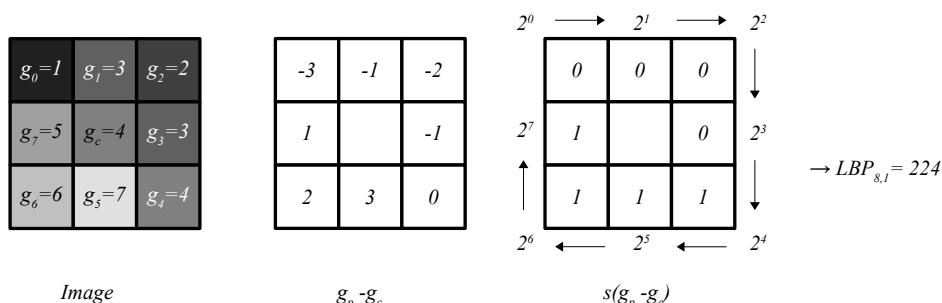


FIGURE 19 – Motifs binaires locaux.

LBP8

8.3 Méthodes fréquentielles

Les textures présentant des motifs répétitifs peuvent être analysées à l'aide de la transformée de Fourier. Elle fournit les informations sur la période de répétition du motif ainsi que sur la direction de répétition.

8.4 Méthodes spatiofréquentielles

Filtres de Gabor. Un filtre de Gabor est un filtre linéaire dont la réponse impulsionnelle est une sinusoïde modulée par une fonction gaussienne.

Le filtre de Gabor peut s'exprimer de la façon suivante :

$$G(x, y, \theta, f) = \exp\left(-\frac{1}{2} \left(\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right)\right) \cos(2\pi f x_\theta)$$

avec $x_\theta = x \cos \theta + y \sin \theta$ et $y_\theta = y \cos \theta - x \sin \theta$. Le filtrage de l'image par ces noyaux est réalisé par convolution.

9 Descripteurs de forme

9.1 Descripteurs classiques

- Aire A , périmètre P , boîte englobante (calculés à partir des composantes connexes).
- Nombre d'angles droits, nombre de coins saillants ou entrants.
- Circularité : A/P^2 , Compacité : $4\pi \times$ circularité.
- Rectangularité = Aire/Aire rectangle exinscrit.
- Eccentricité = longueur axe majeur/ axe mineur
- Nombre d'Euler. Différence entre le nombre de composantes connexes et le nombre de trous.

9.2 Signature et descripteurs de Fourier

Représentation de la forme par une fonction de type $y = f(x)$ (par exemple, distance par rapport au centre de gravité). Une fois la signature calculée, elle peut être caractérisée par les **descripteurs de Fourier**, c'est-à-dire la transformée de Fourier de la signature. La figure 20 montre un exemple de signature définie par la distance des points de contours par rapport au centre de gravité.

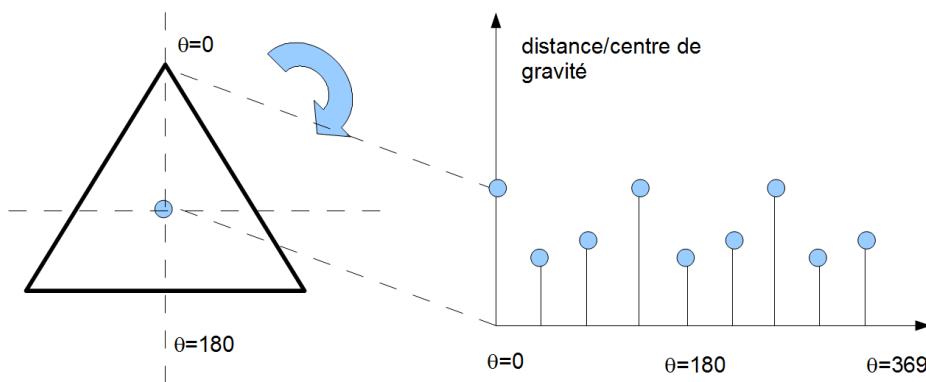


FIGURE 20 – Illustration du calcul de la signature.

9.3 Codage de Freeman

Le codage de Freeman permet de coder un contour en ne codant que les déplacements relatifs. On considère un contour fermé d'épaisseur 1 pixel. Ce contour est parcouru de proche en proche. En chaque point, il y a soit 4 directions (en 4-connexité), soit 8 directions (8-connexité) possibles pour passer au point suivant (voir figure 21). Chacune des directions est codée par 1 chiffre (de 0 à 3 ou de 0 à 7).

Le code obtenu sur la figure 21 est : 11007764606544133442. Une rotation de la représentation aboutit au même résultat. Ainsi, on privilégie une représentation par code minimal, c'est-à-dire que le code subit des rotations jusqu'à atteindre la valeur minimale, ici : 00776460654413344211. Il est possible de compresser le code en comptant les occurrences successives : par exemple 00000077777777 devient 6097 (6 zéros puis 9 septs).

Propriétés utiles pour la reconnaissance

- **Agrandissement** ($\times k$) du contour en répétant k fois chaque descripteur.
- **Rotation** de $k \times 2\pi/n$ (n -connexité) en ajoutant ou retranchant k modulo n à la chaîne initiale.

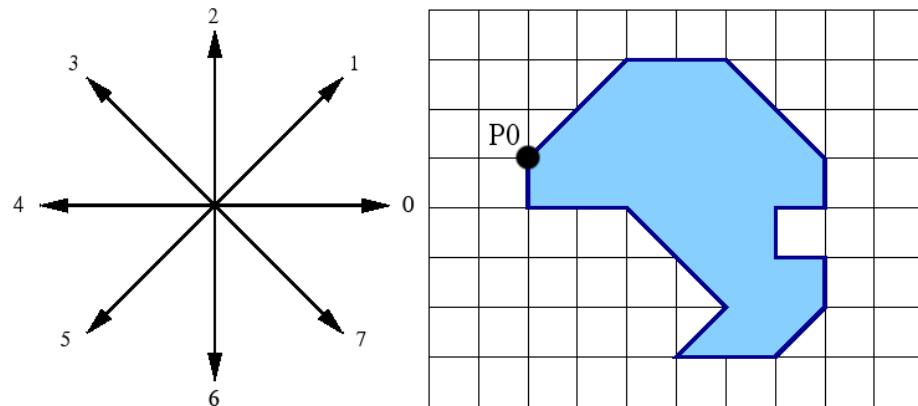


FIGURE 21 – Illustration du codage de Freeman.

- **Longueur du descripteur** : en 4 connexité (=nb. de descripteurs) et en 8 connexité (nb descripteurs pairs + $\sqrt{2}$ nb descripteurs impairs).
- **Inversion d'un chemin** : inversion des descripteurs ($j_{inv} = n/2 + j$ modulo n) puis inversion de la séquence.
- **Simplification d'un chemin** : remplacer p descripteurs consécutifs par des descripteurs équivalents reliant les mêmes points.

9.4 Moments

9.4.1 Moments géométriques

$$M_{pq} = \sum_x \sum_y x^p y^q I(x, y)$$

Aire : M_{00}

Centre d'inertie : $\bar{x} = \frac{M_{10}}{M_{00}}$, $\bar{y} = \frac{M_{01}}{M_{00}}$.

9.4.2 Moments géométriques centrés

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y)$$

9.4.3 Moments invariants de Hu

Les moments invariants de Hu⁵ correspondent à un ensemble de 7 combinaisons d'invariants centrés définis de façon à être invariants vis-à-vis de la translation, ou de l'échelle, ou de la rotation.

5. M. K. Hu, Visual Pattern Recognition by Moment Invariants, IRE Trans. Info. Theory, vol. IT-8, pp.179–187, 1962

9.5 Transformée de Hough

La méthode d'origine a été conçue pour détecter des lignes droites. La méthode se généralise toutefois à tout type de forme paramétrée, comme des cercles, des ellipses, etc. En chaque point (x, y) de l'image, peut passer une infinité de lignes droites dont l'équation peut s'écrire $\rho = y \cos \theta + x \sin \theta$ où θ est l'angle de la droite (dans un repère prédéfini) et ρ la distance de la droite à l'origine du repère. Cette distance ρ est la longueur du segment passant par l'origine et perpendiculaire à la droite. Un espace de votes $H(\theta, \rho)$ est construit. Chaque point (x, y) dans l'image vote pour un ensemble de couples (θ, ρ) en incrémentant la case correspondante dans $H(\theta, \rho)$. Les couples (θ, ρ) ayant obtenu un nombre significatif de votes sont susceptibles de correspondre aux paramètres d'une droite présente dans l'image. La longueur de la ligne est donnée par le nombre de votes (voir figure 22).

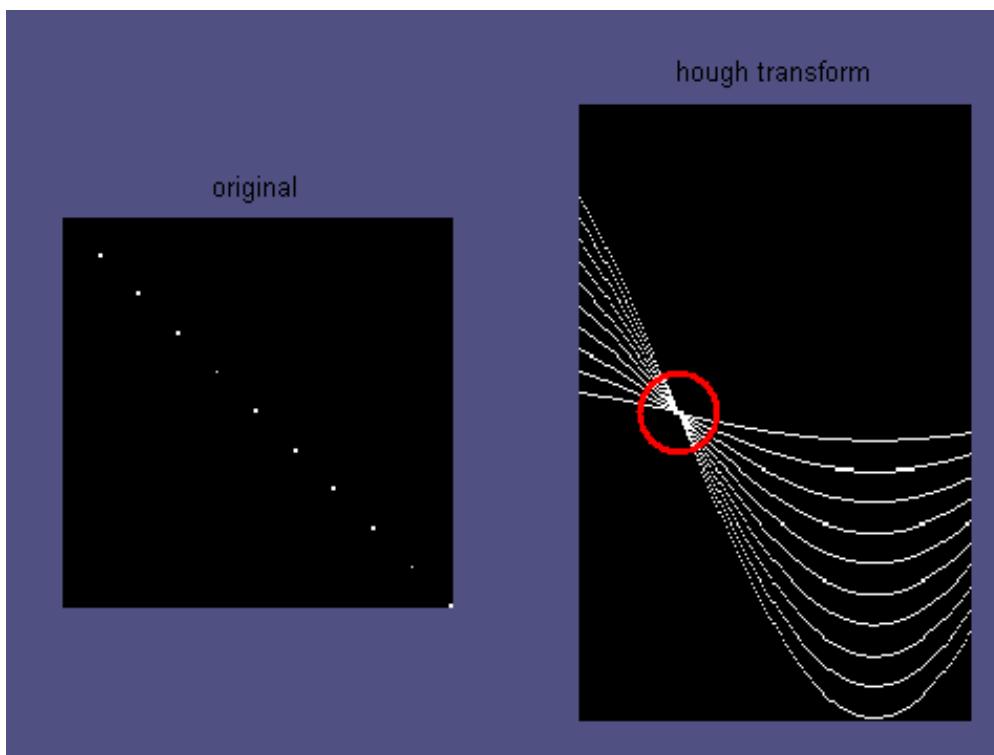


FIGURE 22 – Illustration de la transformée de Hough.

10 Compression d'images

Comprimer une image consiste à la coder de sorte qu'elle prenne moins de place en mémoire. Pour gagner de la place, on exploite la redondance : spatiale liée aux zones de l'images uniformes, temporelle, liée aux zones sans mouvement dans une vidéo. Une façon efficace de comprimer une image consiste à comprimer de façon plus importante les hautes fréquences (contours, textures) car elles sont peu visibles à l'œil.

Le codeur JPEG travaille sur des blocs de taille 8×8 , notés $b(i, j)$ pour $i \in [0, 7]$ et $j \in [0, 7]$. Dans le cas d'images couleur, on effectue une conversion vers l'espace Yc_bc_r .

La compression suit quatre étapes successives :

1. Transformée DCT (passage dans le domaine fréquentiel)
2. Quantification
3. Parcours zig-zag et RLC (Run Length Code)
4. Codage de Huffman

Dans un premier temps, chaque bloc de l'image est transformé dans le domaine fréquentiel à l'aide de la transformée en cosinus DCT (pour *Direct Cosinus Transform*) qui est une variante de la transformée de Fourier. On obtient alors une représentation en bloc $B(i, j)$ de taille 8×8 . La DCT consiste en une somme de fonctions cosinus oscillant à différentes fréquences. Elle permet une représentation en amplitude et en fréquence, plus propice à la compression. En effet, à partir de cette représentation $B(i, j)$, les basses fréquences pourront être facilement conservées (dans la partie en haut à gauche de chaque bloc) et les hautes fréquences sont regroupées en bas à droite. Afin d'appliquer la DCT, les données sont préalablement centrées entre -128 et 127 : $b(i, j) = b(i, j) - 128 \forall i, j$.

Dans un deuxième temps, une quantification est effectuée pour chaque bloc $B(i, j)$ par division par une matrice de quantification $S_Q(i, j) = \text{round}(\frac{S(i, j)}{Q(i, j)})$. Les zones de $S(i, j)$ correspondant aux hautes fréquences vont être quantifiées avec un facteur plus important. Par exemple, pour une image codée en 8 bits on peut avoir ce type de matrice :

$$Q = \begin{pmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{pmatrix}$$

Ainsi, du fait de l'approximation *round* à la valeur entière inférieure, la plupart des coefficients associés aux hautes fréquences sont annulées.

Enfin, chaque bloc est analysé le long d'un parcours en zigzag (comme décrit par la figure 23) et le mot de 64 valeurs qui en découle est codé à l'aide d'un codage RLC (pour *Run Length Code*). Les coefficients associés aux hautes fréquences (et généralement nuls) sont tout simplement oubliés. De ce fait, il n'est pas utile d'utiliser les 64 coefficients pour coder le bloc, mais bien souvent une dizaine de coefficient suffit. Dans un bloc parfaitement uniforme où la DCT ne montre qu'une composante continue, un seul coefficient suffit. On est donc effectivement passé d'un bloc de 64 valeurs redondantes (car identiques) à un seule valeur. Dans le cas où certaines séquences de 0 persistent, on utilise le RLC pour compacter cette séquence : 00000000 devient *huit0* (le huit est codé en utilisant une plage de valeurs différente de [-128, 127] de façon à le distinguer de la valeur 8).

Enfin, le codage de Huffmann permet d'affecter un code différent à chaque coefficient (voir par exemple,<https://lehollandaisvolant.net/science/jpg/>).

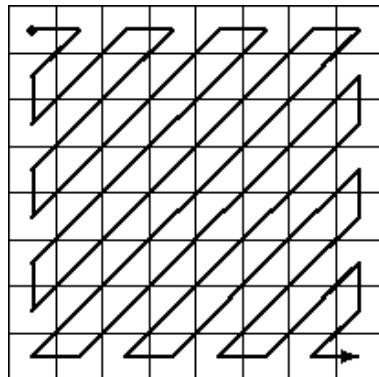


FIGURE 23 – Zig-Zag

11 Quelques références

- Quelques méthodes de filtrage en Traitement d’Image. Maïtine Bergounioux <https://hal.archives-ouvertes.fr/hal-00512280v1/document>
- Digital Image Processing, by Rafael C. Gonzalez and Richard E. Woods, Prentice-Hall, 2002
- The Image Processing Handbook, by John Russ, CRC Press, 1998
- Morphological Image Analysis : Principles and Applications, by Pierre Soille, Springer-Verlag Berlin Heidelberg, 1999
- Analyse d’images : Filtrage et Segmentation, Cocquerez and al., Ed. Dunod, 1995
- Le traitement des images, H. Maître, Hermès Science Publications , 2003
- Les cours d’Antoine Manzanera <https://perso.ensta-paris.fr/manzaner/enseignement.html>