Segmentation d'images : principes

L'analyse d'images a pour but l'extraction de l'information caractéristique contenue dans une image. Le résultat d'une telle analyse s'appelle très souvent la **description structurelle**. Celle-ci peut prendre la forme d'une image ou de toute structure de données permettant une description des entités contenues dans l'image. Par opposition avec la phase d'interprétation, l'analyse tente, dans la mesure du possible, de ne pas prendre en compte le contexte (*i.e.* l'application). Essentiellement, l'analyse de l'image fait appel à la segmentation où l'on va tenter d'associer à chaque pixel de l'image un label en s'appuyant sur l'information portée (niveaux de gris ou couleur), sa distribution spatiale sur le support image, des modèles simples (le plus souvent des modèles géométriques).

La segmentation d'images ainsi définie est un domaine vaste où l'on retrouve de très nombreuses approches.

Toutes ces approches visent à l'extraction des indices visuels. Après de nombreuses années passées à rechercher la méthode optimale, les chercheurs ont compris que la segmentation idéale n'existait pas. On peut même montrer que le problème de la segmentation est le plus souvent un problème mal posé. Etant donnée une image, il existe toujours plusieurs segmentations possibles. Une bonne méthode de segmentation sera donc celle qui permettra d'arriver à une bonne interprétation. Elle devra donc avoir simplifié l'image sans pour autant en avoir trop réduit le contenu. Entre autres, elle devra éviter les choix irréversibles. L'avenir de la segmentation est dans le pilotage aval par l'applicatif.

Liste de mots-clé : Seuillage - Histogramme

Le seuillage a pour objectif de segmenter une image en plusieurs classes en n'utilisant que l'histogramme. On suppose donc que l'information associée à l'image permet à elle seule la segmentation, i.e. qu'une classe est caractérisée par sa distribution de niveaux de gris. A chaque pic de l'histogramme est associée une classe.

Il existe de très nombreuses méthodes de seuillage d'un histogramme. La plupart de ces méthodes s'appliquent correctement si l'histogramme contient réellement des pics séparés. De plus, ces méthodes ont très souvent été développées pour traiter le cas particulier de la segmentation en deux classes (i.e. passage à une image binaire) et leur généralité face aux cas multi-classes n'est que très rarement garantie.

Détection de vallées :

Cette technique est la plus intuitive. On suppose que chaque classe correspond à une gamme distincte de niveaux de gris. L'histogramme est alors m-modal. La position des minima de l'histogramme H permet de fixer les (m-1) seuils nécessaires pour séparer les m classes.

En termes mathématiques, les seuils s_i sont obtenus par

$$H(s_i) = \text{Min} [H(k)] \text{ pour } k \text{ in }]m_i,m_{i+1}[$$

où m_i et m_{i+1} sont les valeurs moyennes (ou les modes) de l'intensité lumineuse dans les classes C_i et C_{i+1}.

Malgré le développement de techniques robustes visant à faciliter la détection des vallées, cette méthode, bien que simple, est très peu appliquée car les histogrammes traités sont le plus souvent bruités et unimodaux.

Minimisation de variance :

La répartition des pixels en N classes est un problème classique de classification. Le choix des seuils s_i permet de détecter m classes auxquels on peut associer taille (t_i), moyenne (m_i) et variance V_i par :

$$t_i = \mathrm{Sum}_{Dj} \left\{ H(j) \right\}$$

$$m_i = \mathrm{Sum}_{Dj} \left\{ j \,.\, H(j) / t_i \right\}$$

$$V_i = \mathrm{Sum}_{Dj} \left\{ (j - m_i)^2 \,.\, H(j) / t_i \right\}$$

où H est l'histogramme normalisé (son intégrale est ramenée à l'unité) et $D_j = [s_{i-1},s_i[$ est la gamme de niveaux de gris correspondant à la classe C_i (par hypothèse, $s_0 = 0$) .

A partir de ces indicateurs statistiques, on peut construire la variance intraclasse totale W par :

$$W = \mathrm{Sum}_i \left\{ t_i . V_i \right\}$$

Le meilleur seuillage dans cette approche correspond à une minimisation de la variance intraclasse (méthode de Fisher). Cette technique est difficilement applicable lorsque le nombre de classes est élevé. En effet, il faut tester exhaustivement tous les (N-1)-uples ($s_1,...,s_{N-1}$) possibles. De plus, il faut que chaque classe ait une taille significative en nombre de niveaux de gris pour que les indicateurs statistiques aient un sens. Dans le cas de la binarisation (N=2), cette méthode est performante.

Plus récemment, Otsu a proposé de réaliser une maximisation de la variance inter-classe qui, dans le cas de la binarisation, s'exprime par

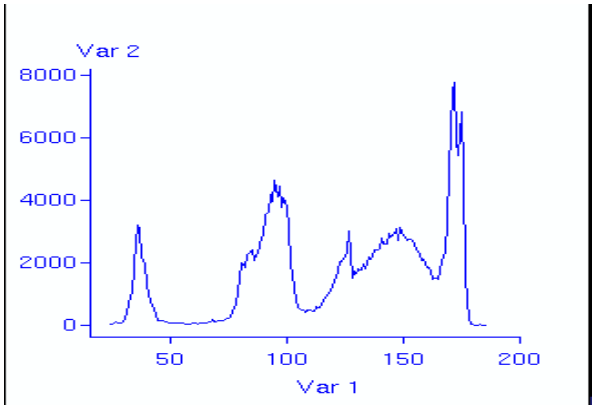
$$B = t_0 \,.\, t_1 \, (m_0 - m_1)^2$$

ce qui est rigoureusement équivalent puisque l'on a la relation :

$$W + B = \mathrm{cste}$$

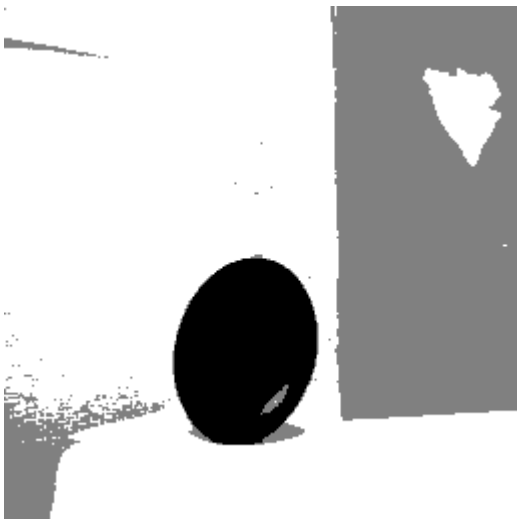
Cependant, la méthode de Otsu est plus intéressante d'un point de vue calculatoire car elle ne nécessite pas de calcul de variances.

D'autres critères statistiques sont utilisables : test de Student, distance de Fisher (pour laquelle il existe un algorithme optimisé), ...



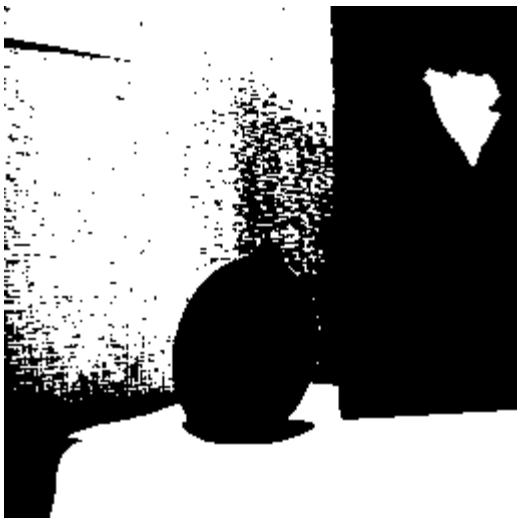
Exemple de seuillage par détection de vallées :

Sur cet exemple, deux seuils on été choisis correspondant aux deux premières vallées : 60 et 110. Si certains objets sont bien détectés, il aurait fallu utiliser plus de seuils dans les niveaux de gris clairs pour mieux mettre en évidence les différentes parties de cette image. C'est le principal inconvénient de beaucoup d'approche du seuillage : combien y-a-t-il a priori de classes à séparer ?



Exemple de seuillage par minimisation de variance :

Sur cet exemple, un seuil a été déterminé par l'algorithme optimal de fisher. Le seuil trouvé est 127. Compte tenu du fait que l'on n'utilise qu'un seul seuil, toutes les composantes de l'image ne peuvent être séparées.



Seuillage entropique :

Le seuillage entropique est une technique dérivée de la théorie de l'information. Les seuils sont déterminés de manière à maximiser l'entropie E résultant du découpage de l'histogramme H en plusieurs classes. En effet, l'entropie mesure la quantité d'information portée par un groupe. Pour un nombre de seuils fixe, on cherche à ce que les classes résultantes portent le maximum d'information.

L'entropie totale est calculée de la manière suivante:

$$E = \mathrm{Sum}_i\ E(C_i)$$

où C_i désigne la classe No i,

$$E(C_i) = -\ \mathrm{Sum}_{Di}\ \{\ p_j\ .\ \log_2(p_j)\ \}$$

où D_i est l'ensemble des niveaux de gris j associés à la classe C_i et p_j la probabilité *a postériori* du niveau de gris j, estimée par

$$p_j = H(j) / \text{taille\_image}$$

La notion d'entropie n'est pas liée à une caractéristique visuelle. C'est pourquoi l'image résultat parait le plus souvent de moins bonne qualité (du point de vue de l'oeil humain) qu'une image obtenue par une des techniques présentées dans ce chapitre.

Méthode du pourcentage :

Dans certains cas, une connaissance *a priori* de la scène, si celle-ci est simple, permet de choisir les seuils. En particulier, si l'image est constituée de deux entités, la connaissance de la fraction surfacique F de l'une des deux phases peut être reliée au seuil s par :

$|d(s) - F| = \mathrm{Min}\ |d(k) - F|$ où d(k) représente l'intégrale normalisée de l'histogramme entre les bornes 1 et k.

Cette technique peut être appliquée au seuillage de pages de caractères si la densité dans une page est connue. Cependant, elle ne s'applique que dans des applications particulières et la généralisation au cas de m classes est difficilement envisageable. Par contre, si la connaissance de F est rarement suffisante, elle peut faciliter la modélisation de l'histogramme.

Bilan :

Les méthodes de segmentation basées sur le choix d'un ou plusieurs seuils sont aujourd'hui utilisées dans des applications très particulières. On les retrouvera en contrôle qualité où la maitrise du milieu ambiant et la simplicité des scènes permet d'avoir de très bonnes conditions d'éclairage. On peut ainsi assurer au mieux l'hypothèse de séparabilité des phases nécessaire au bon fonctionnement de ce type de méthodes.

C'est dans ce type d'applications que l'on exploite le mieux les deux grands avantages du seuillage : la simplicité et la rapidité. Ces deux caractéristiques sont liées. En effet, la simplicité des algorithmes permet souvent la réalisation d'opérateurs cablés permettant la rapidité nécessaire dans le contexte du temps réel qui est le contexte habituel des applications industrielles.

Bien que nécessitant des paramètres, les techniques de seuillage peuvent fonctionner en mode autonome. Il est possible d'affiner un seuil par comparaison entre l'image initiale et l'image seuillée ou bien en rajoutant des contraintes sur la qualité (à définir) de l'image seuillée. Cette mesure peut être interprétée en terme d'évaluation de l'erreur de classification.

L'histogramme étant une fonction trop globale, on peut envisager un seuillage qui ne soit pas fixe. Pour cela, l'image est décomposée en sous-images. Sur chacune d'elles, une technique classique de seuillage est appliquée. Cependant, ce processus risque d'introduire des discontinuités. De plus, si les sous-images sont trop petites, les histogrammes ne contiendront pas assez de valeurs pour être statistiquement exploitables. Cette technique est à réserver à des configurations particulières (dérive lumineuse par exemple).

Cependant, ces méthodes ne sont pas suffisantes pour la plupart des applications où la complexité de l'information contenue dans l'image ne peut être résumée par l'histogramme des niveaux de gris sans trop de pertes d'information.

Du pixel au contour :

La recherche des contours dans une image numérique est un des problèmes les plus étudiés depuis l'origine des travaux sur l'imagerie numérique. Ceci est en grande partie dû à la nature très intuitive du contour qui apparait très naturellement comme l'indice visuel idéal dans la plus grande partie des situations.

Très schématiquement, les contours sont les lieux de variations significatives de l'information niveaux de gris (il y a très peu de travaux sur les contours dans les images couleurs ou multispectrales). Dans cette approche, on suppose que l'image est une mosaïque de régions parfaitement homogènes. C'est à dire que les contours recherchés sont de type **créneaux**. De plus, la transition étant stricte, un contour doit être une chaîne de pixels d'épaisseur 1. Cette restriction sur la nature du contour a été imposée dans un premier temps pour des raisons de formalisation mathématique. Il est possible de construire des processus capables d'extraire d'autres types de contours comme par exemple des vallées ou des toits. Cependant, il n'existe pas à l'heure actuelle de processus complet et général qui pourrait extraire tous les types de contour.

La notion de contour étant reliée à celle de variation, il est évident qu'une telle définition nous amène tout naturellement vers une évaluation de la variation en chaque pixel. Une variation existera si le gradient est localement maximum ou si la dérivée seconde (à définir dans un espace bi-dimensionnel) présente un passage par zéro. Les principaux algorithmes connus (Sobel, Prewitt, Kirsh, Canny, Dériche, ...) se focalisent sur ce premier aspect du contour.

Il existe moins de travaux sur la formalisation de la deuxième partie consistant à passer d'une mesure locale de variations à des chaines de points d'épaisseur 1. C'est pourtant cette deuxième partie qui fait souvent la différence et la qualité visuelle d'un résultat.

Calcul d'un gradient :

Le gradient, en un pixel d'une image numérique, est un vecteur caractérisé par son amplitude et sa direction. L'amplitude est directement liée à la quantité de variation locale des niveaux de gris. La direction du gradient est orthogonale à la frontière qui passe au point considéré.
La méthode la plus simple pour estimer un gradient est donc de faire un calcul de variation monodimensionnelle, *i.e.* en ayant choisi une direction donnée. On a alors le schéma suivant :

$$G_d(x,y) = (I * W_d)(x,y)$$

où W_d désigne l'opérateur de dérivation dans la direction d et * le produit de convolution.

$$G_d(x,y) = \mathrm{Sum}_{i=-m,+m}\ \mathrm{Sum}_{j=-n,+n}\ I(x+i,y+j)\ .\ W_d(i,j)$$

Dans cette version discrète, la taille de cet opérateur est donnée par le couple (m,n). Sauf cas très particulier, on utilise toujours m = n.

Il existe de très nombreux opérateurs différents (Roberts, Sobel, Prewitt, Kirsch, ...) qui ont globalement les mêmes propriétés.

Deux ou plusieurs directions...

Le gradient étant un vecteur, l'approche la plus classique pour estimer le gradient consiste à choisir deux directions privilégiées (naturellement celles associées au maillage, *i.e.* ligne et colonne) orthogonales, sur lesquelles on projette le gradient. A partir de deux calculs identiques à celui

présenté ci-dessus, on peut donc obtenir une connaissance parfaite du gradient :

$$G(x,y) = \left(G_X(x,y), G_Y(x,y)\right) = \left((I * W_X)(x,y), (I * W_Y)(x,y)\right)$$

L'amplitude du gradient s'obtient alors par l'une des formules suivantes :

$$m(x,y) = (G_X(x,y)^2 + G_Y(x,y)^2)^{1/2}$$
$$m(x,y) = |G_X(x,y)| + |G_Y(x,y)|$$
$$m(x,y) = \mathrm{Max}(|G_X(x,y)| + |G_Y(x,y)|)$$

Et la direction du gradient est donnée par:

$$d(x,y) = \mathrm{Arctg}(\, G_Y(x,y) / G_X(x,y))$$

Cette approche présente comme principal inconvénient le fait que la direction peut prendre n'importe quelle valeur réelle. Ceci n'est pas en accord avec la nature discrète d'une image. Que représente une frontière orientée à 17° sur une grille ? C'est pourquoi on peut adopter un schéma différent adapté à la nature discrète de l'image. Il s'agit alors de calculer le gradient, non plus dans deux directions, mais dans toutes les directions possibles de l'image : 0°, 45°, 90°, 135°. On peut se contenter de ces 4 directions. On a alors :

$$m(x,y) = \mathrm{Max}_{d = 0°, 45°, 90°, 135°}\, m_d(x,y)$$
$$d(x,y) = \arg \mathrm{Max}_d\, m_d\,(x,y)$$

Là encore, on peut imaginer plusieurs types de masques (Sobel, Prewitt, Kirsch, ...)

Et la texture...

La texture est très souvent considérée comme un élément perturbateur car caractérisée par des transitions mais peu intéressante en termes de contours d'objets. Il existe donc des travaux visant à calculer les transitions hors textures. Le plus souvent, on ramène les calculs de variations entre valeurs de pixels à des calculs entre valeurs moyennes sur des fenètres centrées sur ces pixels. C'est le cas de l'opérateur de Rosenfeld.

Sur la taille de l'opérateur...

Les masques qui sont détaillés ici sont des masques m=n=1 (3 x 3). Il s'agit des plus couramment utilisés. Cependant, ils ne sont qu'un cas particulier. Asfar a proposé une généralisation du masque de Sobel pour une taille de voisinage quelconque. Cependant, comment peut-on savoir quelle taille choisir pour un opérateur ? Nous donnons ici quelques règles simples :

- Plus le masque est grand, moins le gradient est sensible au bruit.
- Plus le masque est grand, plus le temps de calcul est élevé.
- Plus le masque est grand, moins bonne est la localisation des contours.

Bien sûr, augmenter la taille du masque présente à la fois des avantages et des inconvénients, et il faut donc réaliser un compromis. De plus, pour une scène de complexité normale, il existe souvent des contours de natures très différentes qui nécessiteraient l'emploi de plusieurs tailles de masques. C'est l'approche qui est le plus souvent préconisée. Malheureusement, il n'existe pas de théorie formalisée sur le mixage d'informations provenant de plusieurs masques et seulement des "recettes" algorithmiques.

Detection de contour

Modèle de Canny-Dériche : Résumé

Canny a proposé en 1983 une étude théorique de la détection de contour. Son étude s'est limitée au cas de la dimension 1, c'est à dire la détection des variations dans un signal bruité.

Il a le premier formalisé trois critères que doit valider un détecteur de contour :

- Détection : robustesse au bruit
- Localisation : précision de la localisation du point contour
- Unicité : une seule réponse par contour

A chaque critère est associé une formule mathématique. La maximisation de ces critères conduit à la résolution d'une équation différentielle dont la solution est le filtre f, qui permet la détection du contour, i.e. la position du contour correspond à :

$$\max (I*f) (x)$$

En fixant des conditions initiales, Canny a montré que cette solution générale pouvait être approximée par :

$$f(x) = -(x / \boldsymbol{tau}^2)\, e^{-(x^2 / 2\boldsymbol{tau}^2)}$$

C'est à dire la dérivée du filtre gaussien.

Le passage à un espace à 2 dimensions (une image) se fait alors simplement car le filtre gaussien étant séparable, on a :

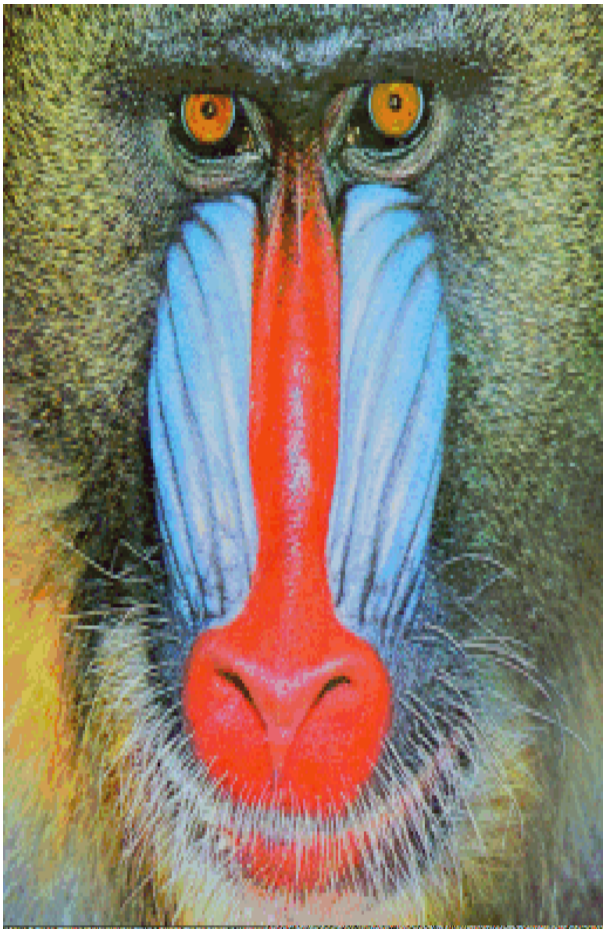
$$f(x,y) = f(x) \,.\, f(y)$$

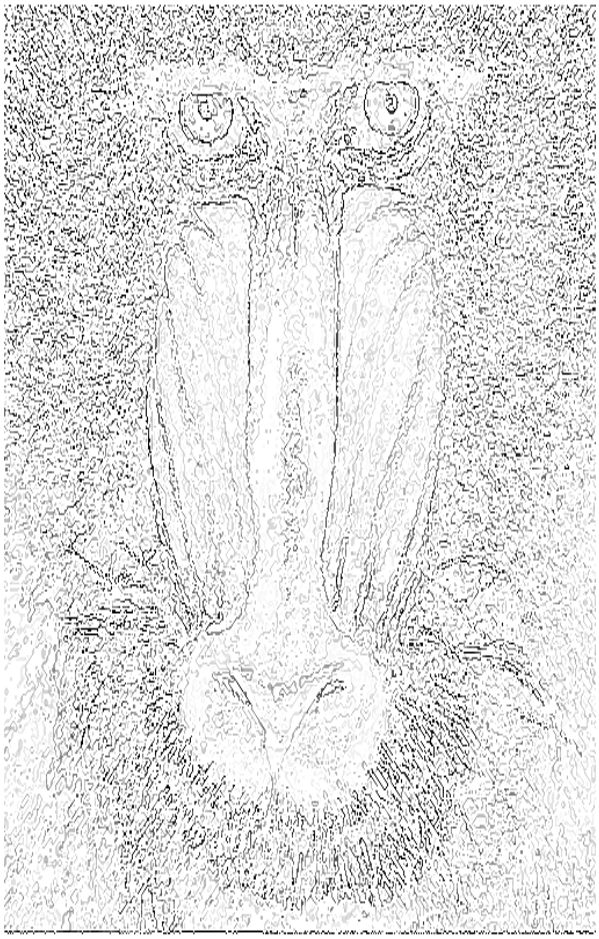
Le calcul du gradient sur une image se ramène donc à deux balayages (un en ligne et un en colonne).

Partant d'autres conditions initiales, Dériche a proposé un filtre différent dont la forme simplifiée est :

$$f(x) = (s\ x)\, e^{-\boldsymbol{alpha}|x|}$$

Le principal avantage de cet opérateur est l'optimisation que l'on peut faire de son implémentation.





Contour ou contraste : Résumé

La notion de contour est très utilisée en analyse d'images. Pourtant, elle est trop souvent associée à celle de transition entre deux niveaux de gris constants (modèle du créneau). Bien sûr, il y a eu des essais pour tenter de définir des contours généralisés (en toit, en rampe, ...) mais sans un réel succès car cela n'a jamais débouché sur un détecteur unique.

C'est cette référence à un modèle mathématique géométrique qui fait à la fois la force et la faiblesse du contour. Par opposition, on peut aussi rechercher une entité plus pauvre que le contour, donc plus robuste : le contraste.

Le **contraste** est défini par la variation entre le niveau de gris local et le niveau représentatif d'un voisinage plus global. Le plus souvent, cela se traduit par un ratio entre le niveau de gris en un point et le niveau de gris moyen sur une fenêtre centrée sur ce point.

$$C(p) = I(p) / Moy\{I(q) : q \text{ dans } V(p)\}$$

ou sous sa forme normalisée :

$$Cn(p) = (I(p) - M(p)) / (I(p)+M(p)) \text{ avec } M(p) = Moy\{I(q) : q \text{ dans } V(p)\}$$

C(p) pose des problèmes si le voisinage a une moyenne nulle (mais si p appartient à son voisinage, on a alors 0/0 car I(p) = 0). Cn(p) est normalisé sur [-1,1].

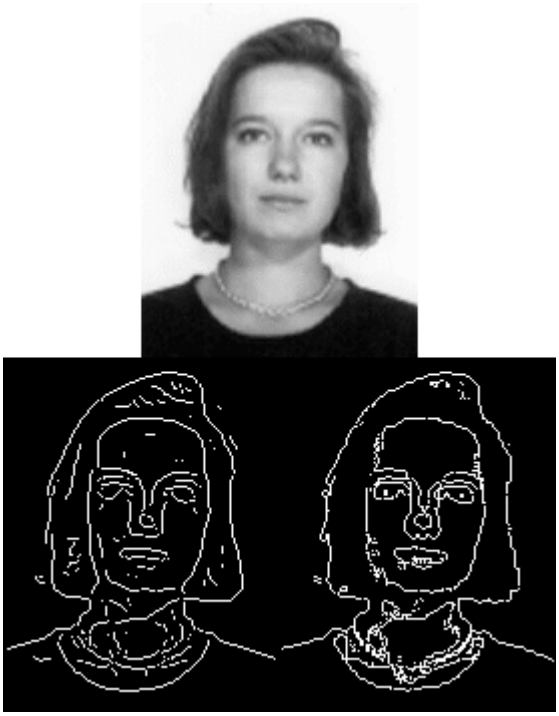
Cette mesure est surtout utilisée pour des réhaussements. On peut également en déduire des images binaires où ne subsiste que le contraste.

La principale différence entre contraste et contour est bien sûr le fait qu'il n'y a pas de prise en compte de structure dans le contraste, on n'a donc pas de contrainte du type de celles utilisées en

détection de contour (épaisseur 1 pixel, forte continuité, ...). Les images contrastes apparaissent donc beaucoup plus brutes.

Contour ou Contraste : Exemple

Que préférez-vous : le contour (à gauche, extrait par l'opérateur de Canny-Dériche) ou bien le



contraste (à droite) ?

Modèle région :

La segmentation d'une image vis à vis d'un critère d'homogénéité H peut s'exprimer de la manière suivante :
La segmentation d'une image I en regard du critère H est une partition de l'image I en régions homogènes X_1, ..., X_n telles que :

1. $\text{Union}_{i=1,\ldots,n}\ X_i = I$
2. Pour tout i, X_i est connexe
3. Pour tout i, $H[X_i]$ est vrai
4. Pour tout couple (X_i,X_j) de régions voisines, $H[X_i,X_j]$ est faux

Cette définition conduit à deux remarques très importantes. Tout d'abord, une segmentation dépend du critère employé. Le choix du critère est donc primordial. Ensuite, la décomposition obtenue n'est pas unique. Pour un critère donné, il existe plusieurs solutions.

Split_and_merge :

L'algorithme *Split_and_merge* a été proposé par Horowitz et Pavlidis en 1974. Le processus est décomposé en deux étapes. L'image initiale peut être une première partition résultant d'une analyse grossière ou bien l'image brute. Dans la première étape, ou division, on analyse individuellement chaque région X_i. Si celle-ci ne vérifie pas le critère d'homogénéité, alors on divise cette région en blocs (le plus généralement en 4 quadrants) et l'on réitère le processus sur chaque sous-région prise individuellement. On peut tout à fait initier le processus en considérant que la première région est composée de toute l'image.

Dans la deuxième étape, ou réunion, on étudie tous les couples de régions voisines (X_k,X_i). Si l'union de ces deux régions vérifie le critère d'homogénéité, alors, on fusionne les régions. La principale difficulté de cette approche réside bien sûr dans le parcours de l'ensemble de tous les couples de régions voisines.

Pyramides adaptatives et stochastiques

Les techniques dites de pyramides irrégulières sont utilisées en segmentation d'images pour partitionner une image en régions par un procédé de type *bottom-up region growing*. A l'état initial, chaque pixel de l'image à segmenter est une région. Ces régions sont les noeuds d'un graphe dont les arêtes sont les liens de voisinage entre régions. Le processus est itératif et à chaque itération, il tente de supprimer le maximum de régions par agglomération (*i.e.* réduction du graphe). Pour atteindre cet objectif, il est nécessaire de définir des règles de fusion. Selon les règles, on obtient différentes techniques dont les plus connues sont :

- la pyramide stochastique
- la pyramide adaptative
- la pyramide duale

Pyramides adaptatives et stochastiques : détail

Règles de fusion

A l'itération *k*, l'image I_k est un graphe dont les noeuds sont les régions $R_{k,i}$ ($i = 1, ..., n_k$) et les arêtes, les liens de voisinage entre ces régions.

La construction de l'image à l'itération *k*+1 a pour but de regrouper les régions voisines afin d'avoir $n_{k+1} < n_k$. Le principal problème, déjà rencontré dans la phase *merge* de l'algorithme *split & merge*, réside dans le choix de l'ordre dans lequel on procède à ces regroupements car ceux ci ne sont bien sûr pas tous indépendants.

Les techniques de pyramides irrégulières ont eu pour principal mérite de résoudre le dilemme grâce à la formalisation du problème de la fusion en terme de réduction de graphe. Plus exactement, le problème du choix des fusions se ramène à la recherche d'un stable dans le graphe. Ceci s'opère en deux étapes :

1. Sélection d'un sous ensemble de noeuds du graphe initial&bnsp;: les régions survivantes

2. Affectation de toutes les régions non survivantes à une région survivante en conservant la cohérence du graphe

On doit à P. Meer un algorithme parallèle très efficace pour atteindre ce double objectif. Afin d'obtenir une réduction importante du nombre de régions entre deux itérations successives, P. Meer a proposé d'imposer la contrainte suivante :

> C_1 : Deux régions voisines (*i.e.* en relation dans le graphe) ne peuvent survivre toutes les deux

Afin de faciliter l'affectation des régions non survivantes, il est nécessaire que cette décimation ne soit pas trop brutale. Pour cela, on impose une deuxième contrainte :

> C_2 : Toute région non survivante possède, parmi ses régions voisines, au moins une région qui va survivre

Pour valider ces deux contraintes, on applique l'algorithme suivant :
A chaque région R, on associe deux variables binaires p et q qui indiquent si la région valide les contraintes C_1 et C_2.
Etape 1 : Pour toute région R, $p(R) = 1$ ssi $x(R) >= [l(i)\ x(i)]$
La variable x, associée à la région R, traduit son aptitude *a priori* à être retenue dans le graphe suivant. Les méthodes se différencient sur la nature de cette variable ainsi que sur l'emploi du paramètre $l(i)$.

Exemple :

->

Graphe initial avec les valeurs de x(R) **Régions suivantes après la première étape (notée en noir)**

Cette première étape assure que la contrainte C_1 soit validée. Un processus itératif permet de valider la contrainte C_2.

Etape n :

a. $q_n(R) = 1$ ssi $P_{n-1}(R) = 0$
b. $p_n(R) = 1$ ssi $x(R) >= [l(i)\ x(i)]$ tel que $q_n(i) = 1$

La variable $q_n(R)$ sera positionnée à 1 si la région R ne peut valider la contrainte C_2. La recherche de nouveaux minima locaux est alors restreinte à ces seules régions. Ce processus itératif est convergent.

Exemple : représentation des étapes 1, 2a et 2b

Dans cet exemple, la région associée à la valeur 5 est déclarée survivante (alors qu'elle n'est pas un minimum local) afin de valider la contrainte C_2.

La phase de fusion concerne alors les régions R telles que $p(R) = 0$ à l'issue de la convergence du processus de décimation. Comme une telle région peut avoir plusieurs régions voisines survivantes, un critère de choix est nécessaire.

Pyramide stochastique (P. Meer, 1986)

Dans ce modèle initial, les régions survivantes sont choisies aléatoirement, *i.e.* $x(R)$ est la réalisation d'une variable aléatoire uniforme. En cas de présence de plusieurs régions survivantes dans le voisinage de la région à fusionner, un tirage aléatoire est mis en oeuvre.

Pour une image de taille $2^N \times 2^N$, le nombre de graphes intermédiaires (I_k) est de l'ordre de N+1. De plus, on peut montrer que le ratio de décimation $I_k)/I_k)$ est en moyenne égal à 5,44 (± 0,067).

Si le processus converge vite, l'emploi de variables aléatoires rend le résultat indépendant du contenu de l'image. En ce sens, cette méthode est beaucoup trop extrémiste.
Grâce aux paramètres $l(i)$, on peut cependant favoriser l'émergence de régions survivantes dans des directions privilégiées.

Pyramide adaptative (Jolion et Montanvert, 1989)

Afin de relier le processus des décimations au contenu de l'image, on prend pour la variable *x* une mesure de variation, *i.e.* la variance. Ceci revient à favoriser l'émergence des régions homogènes. De plus, afin de ne pas converger vers un graphe où une seule région représente toute l'image, on autorise la neutralisation de régions non survivantes dans les graphes intermédiaires. Une région sera neutralisée si et seulement si elle n'est pas survivante en regard d'un critère de contraste :

> **si** R est non survivante **et si** R' est la plus similaire de ses régions voisines survivantes
> **alors**
> **si** contraste(R,R') > seuil
> **alors** R est neutralisée
> **sinon** R fusionne avec R'

La similarité et le contraste sont approximés par la différence entre les niveaux de gris moyen des régions (d'autres expressions sont bien sûr possibles).

La neutralisation introduit un ralentissement dans le processus de réduction du graphe. Cependant, le résultat est de bien meilleur qualité.