

Traitement d'images
Traitement d'images
et
et
vision.
VISION

Tables des matières

Sujets	Page
<u>Types d'images et leurs applications.....</u>	<u>3</u>
<u>Images de transmission.....</u>	<u>3</u>
<u>Images soniques.....</u>	<u>3</u>
<u>Images radars.....</u>	<u>4</u>
<u>Images « pression ».....</u>	<u>4</u>
<u>Images « distances à l'objet ».....</u>	<u>4</u>
<u>Images bi-niveaux (Noir et blanc).....</u>	<u>4</u>
<u>Utilité des images bi-niveaux.....</u>	<u>5</u>
<u>Traitement de l'image.....</u>	<u>6</u>
<u>Convolution spatiale (algorithme de traitement d'image).....</u>	<u>7</u>
<u>Détection des contours d'une image.....</u>	<u>13</u>
<u>Conversion de couleurs à ton de gris.....</u>	<u>14</u>
<u>Seuil de filtrage (Binarisation).....</u>	<u>15</u>
<u>Analyse d'image.....</u>	<u>18</u>
<u>Identification des régions d'une image.....</u>	<u>18</u>
<u>Propriétés mesurables des régions.....</u>	<u>22</u>
<u>Aire.....</u>	<u>22</u>
<u>Périmètre.....</u>	<u>22</u>
<u>Centre de masse.....</u>	<u>25</u>
<u>Circularité.....</u>	<u>25</u>
<u>Rectangularité.....</u>	<u>26</u>

Types d'images et leurs applications

La plupart des images sont formées de la même manière : de la lumière est réfléchi sur un objet. Cette lumière réfléchi est ensuite convertie par un mécanisme en une forme digitale. Cette façon de faire n'est cependant pas la seule qui permet de produire une image et, en fait, la lumière n'est pas essentielle à la production d'une image.

Le son et la pression par exemple peuvent devenir des sources qui ultimement produiront une image. Dans certains cas également, les images sont produites par absorption de la lumière plutôt que par la réflexion. Voici quelques exemples :

Images de transmission

Elles sont produites lorsque la lumière passe au travers un objet. L'exemple le plus courant de ce type d'image est le « rayon-X ». La radiation passe au travers l'objet : plus celui-ci est dense moins la radiation aura tendance à passer. Ainsi, les régions plus denses comme les os seront plus foncées tandis que les régions moins denses seront plutôt pâles.

Images soniques

Elles sont produites par la réflexion des ondes de son sur un objet. Bien que des fréquences de son élevées soient utilisées pour améliorer la résolution de l'image, elles demeurent difficiles à interpréter. L'appareil qui permet de faire des ultrasons en médecine en est un exemple. L'image est générée en convertissant électroniquement les intensités du son perçue par un microphone en des intensités de lumières. Les images soniques sont également utilisés pour déterminer la distance des objets (sous-marins).

Images radars

Elles sont produites de plusieurs façons. La vue traditionnelle du radar avec le faisceau lumineux est en une mais les systèmes plus modernes permettent de produire des images beaucoup plus précises. Les images radars sont souvent produites par les satellites météorologiques ainsi que par les satellites artificiels orbitant des planètes ou des lunes.

Images « pression »

Elles sont habituellement obtenues au moyen de capteurs qui permettent de transformer la pression obtenue en intensité de lumière. On peut ainsi obtenir une idée générale de l'objet que l'on presse. Certains capteurs de ce type sont utilisés dans des mains « robotiques » pour ainsi simuler le sens du touché.

Images « distances à l'objet »

Elles consistent en un tableau contenant la mesure de la distance d'un objet dans un scène. Elles peuvent être produites par « sonar » ou par détecteur laser. Encore une fois, l'application de ce type d'images se retrouve en robotique. Les portions plus foncées représentent des distances plus grandes alors que les portions plus pâles sont des distances plus courtes.

Images bi-niveaux (Noir et blanc)

Les images bi-niveaux ne contiennent que deux tons de gris : un blanc, l'autre noir. Les images de ce type peuvent être représentées en utilisant seulement un bit par pixel et, si désiré, peuvent être compactés en une suite de 8 bits pour former un octet. La forme résultante, appelée **bitmap** peut ainsi être directement affichée sur un écran. Généralement, les images bi-niveaux ont été produites en compressant l'intervalle de ton de gris jusqu'à ce qu'il y en ait que deux. Le résultat de cette image contient ainsi moins d'informations que l'image originale et, par conséquent, cette opération appelée **thresholding** n'est réalisée que lorsque nous savons à l'avance qu'elle ne détruira pas l'image originale.

Les images dans lesquelles l'information vitale est composée de lignes peuvent être transformées en images bi-niveaux. Ce que l'on veut obtenir, c'est la direction et la longueur de la ligne et sa relation avec les autres lignes formant l'image et non pas son niveau de gris.



Un autre type d'image bi-niveaux est l'image dite « classifiée » dans laquelle les portions importantes de l'image ont été identifiées et transformées en noir alors que le fond de l'image a été transformée en blanc ou vice-versa. Ceci nous permet d'avoir une base pour la reconnaissance de formes. De plus, les images bi-niveaux sont faciles à afficher sur plusieurs types de périphériques.

Utilité des images bi-niveaux

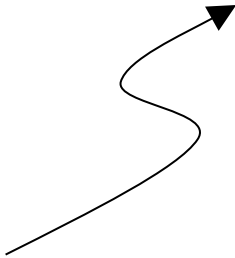
On rencontre souvent les images bi-niveaux dans les images représentant du caractère imprimé (un texte écrit par exemple). Les caractères tapés à la machine ou écrit à la main sont facilement lisibles par un humain et ce, même s'il n'y a que 2 tons de gris utilisés. Ce n'est donc pas la relation entre les niveaux de gris qui nous intéresse ici mais plutôt la forme de l'objet (dans le cas précédent, les caractères) qui contiennent l'information nécessaire. Ceci est la propriété commune aux images bi-niveaux et les images de tons de gris ayant cette propriété peuvent être converties en format bi-niveaux sans perdre l'information qu'elles contiennent.

Les images formées par des lignes sont de bonnes candidates à être transformées en images bi-niveaux. Les graphes de même que les cartes routières sont d'autres exemples d'images de lignes. La représentation bi-niveaux d'une carte routière nous permet d'identifier les routes, les rivières ainsi que les villes et pourrait nous permettre de calculer les distances entre deux villes pour certaines routes empruntées. Par contre, certaines cartes utilisent des tons de gris ou de la couleur pour identifier les zones ombragées et sans ces informations, il pourrait s'avérer difficile d'identifier les lacs, les montagnes, etc.

Comme vous allez le constater, le traitement d'image est souvent réalisé par des opérations arithmétiques sur la valeur des pixels de l'image.

En conclusion, les images bi-niveaux comportent certains avantages :

- Elles sont facilement identifiables,
- Elles permettent généralement des plus grandes vitesses de traitement et
- Elles sont simples à traiter au niveau informatique.



Traitement de l'image

Au niveau informatique, une image peut être représentée par un tableau à 2 dimensions d'éléments que l'on nomme pixel.

Définition d'une image d'un point de vue informatique.

Ce sont ces pixels qui, lorsqu'ils sont affichés très proche l'un des autres, nous donnent l'impression d'une scène continue et constituée d'objets. Chaque pixel est représenté par un ou plusieurs bits. Dans une image "Noir et blanc", chaque pixel est représenté par un seul bit. Dans les images possédants 256 couleurs, 8 bits (1 octet) sont nécessaires pour représenter l'image. Les meilleures images, en termes de qualité, utilisent 3 octets par pixel, un octet pour chaque composante de la couleur primaire (rouge, vert et bleu). Ces images sont souvent appelées "true colors" parce qu'elles ne sont pas approximées.

Les images composées de 256 couleurs sont basées sur une palette de couleur. Le programme qui a créé l'image a sélectionné les 256 couleurs qui décrivent le mieux l'image et les a sauvegardés dans une palette qui est elle-même contenu dans la sauvegarde de l'image.

Le traitement d'image proprement dit consiste à appliquer un changement à l'image en réalisant des opérations sur les pixels de l'image.

Définition du traitement d'image

Parce que les images sont ni plus ni moins que des tableaux 2 dimensions d'entiers, leur traitement est souvent réalisé en appliquant des opérations arithmétiques sur ces entiers.

Convolution spatiale (algorithme de traitement d'image)

Nous allons, dans cette section, développer quelques algorithmes relatifs au traitement de l'image. Nous verrons l'implantation d'algorithmes classiques dans ce domaine. Les effets de filtre les plus rencontrés telles l'adoucissement « blur », le renforcement « sharpen », l'effet relief « emboss » ne sont que quelques exemples de filtres. Ces filtres sont appliqués à l'image originale en utilisant une matrice de $N \times N$ éléments. Cette opération, qui consiste à traiter les pixels à l'aide de cette matrice, se nomme **convolution spatiale**.

Effet d'adoucissement (blurring, smoothing)

C'est une des opérations les plus communes dans le traitement des images. Les images traitées avec l'adoucissement comporte moins de changement abrupte que dans l'image originale et a une apparence de verre semi-transparent.

L'opération d'adoucissement est équivalente à une opération de filtrage "basse fréquence". On filtre l'image pour ne laissez passer que les basses fréquences. L'analogie est la même que si on utilisait un égalisateur "equalizer" pour couper les sons à haute fréquence dans votre chaîne stéréo.

Sur une image, on peut également couper les hautes fréquences. Mais que sont ces hautes fréquences dans une image ?

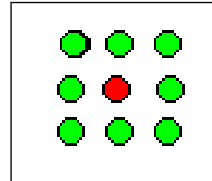
Ce sont les portions de l'image où les changements d'intensité sont très abrupts. Ce sont ces portions qui seront le plus affectées par le filtre.

Propriétés de l'image qui est filtrée par l'adoucissement.
--

Principe

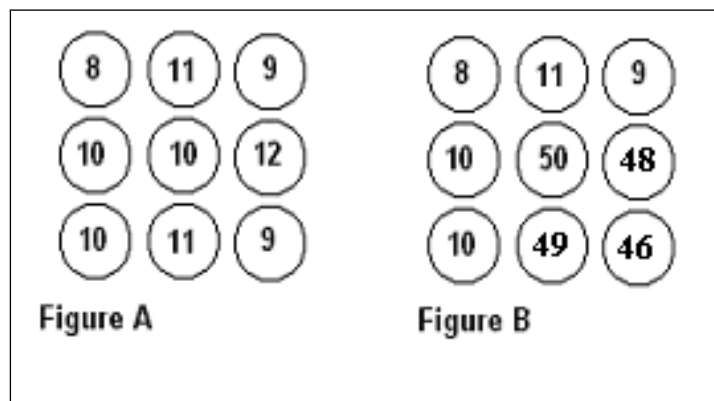
Vous devez réduire les grandes différences d'intensité entre les pixels adjacents. Prenons un pixel central et ses 8 voisins. Voir figure ci-dessous.

Ce bloc contient le pixel à adoucir (celui du milieu) et ses 8 pixels voisins. Supposons également que le pixel du centre est de couleur ROUGE et que ses voisins sont de couleurs VERT. Le pixel rouge est drastiquement différent et pour qu'il devienne "embrouillé", il doit s'approcher de la moyenne des autres pixels qu'il l'entoure.



Exemple:

Pour comprendre que s'est bel et bien le changement d'intensité qui provoque un adoucissement prononcé ou non, prenons l'exemple des 9 pixels ci-dessous:



Dans le cas de la figure A, la moyenne donne $(8 + 11 + 9 + 10 + 10 + 12 + 10 + 11 + 9) / 9 = 10$

On peut dire que la différence, en pourcentage, entre la valeur du pixel central et de la moyenne est de 0%.

Dans le cas de la figure B, la moyenne donne $(8 + 11 + 9 + 10 + 50 + 48 + 10 + 49 + 46) / 9 = 26.78$

Le valeur du pixel a diminué de près de 46%. En effet, $(50 - 26.78) / 50 * 100 = 46.4\%$

Facteurs qui influencent l'adoucissement

Un autre facteur affectant l'adoucissement est le nombre de pixel voisin utilisé pour calculer les valeurs. Dans l'exemple ci-dessus, nous avons utilisé un masque de 3 x 3. On pourrait utiliser un masque de 5x5 pour effectuer un adoucissement encore plus marqué.

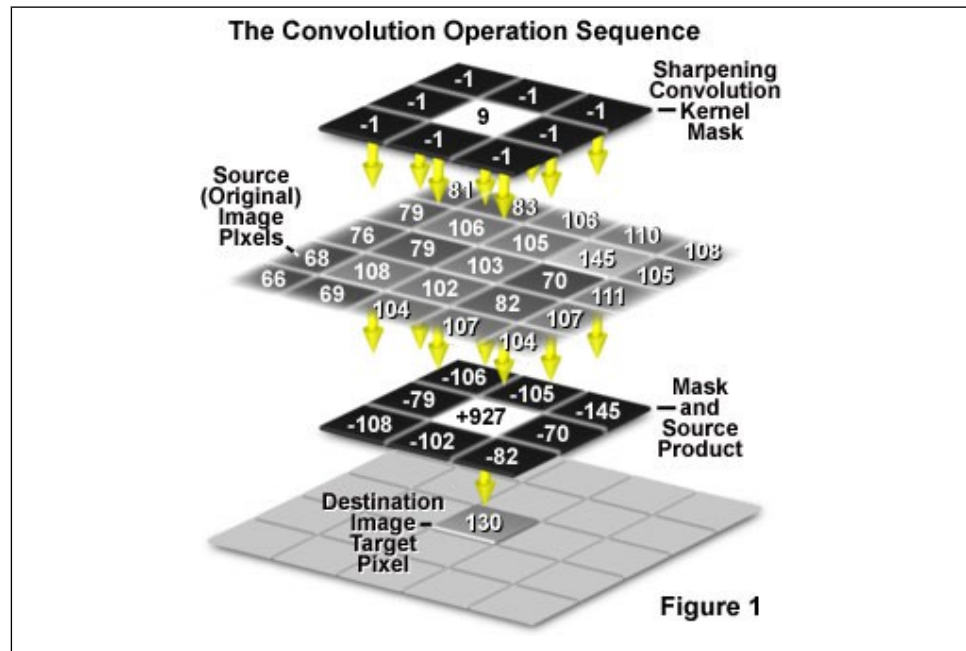
Attention:

Plus le masque couvre une grande quantité de pixel, plus l'adoucissement donnera un résultat qui fera en sorte de perdre les informations de l'image.

L'effet d'adoucissement (blur) exprimé sous forme de matrice de convolution

Qu'est-ce qu'une convolution ?

C'est la procédure qui consiste à appliquer une matrice de N x N aux pixels d'une image. Graphiquement, la convolution peut se représenter de la façon suivante :



Le filtre « blur » peut être exprimé sous une forme plus compacte et représente une matrice de $N \times N$ où N est le nombre de rangée et colonne qui font partie de la matrice. Cette matrice se nomme « matrice de convolution ».

Ainsi, dans l'exemple précédent, l'effet « blur » est exprimé de la façon suivante :

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

→ **Matrice de convolution**

Ou encore

1	1	1
1	1	1
1	1	1

 / **9**

Effet de renforcement (sharpen)

Le filtre de renforcement est le contraire de la technique utilisée pour le filtre de « blur ». Il consiste à augmenter les endroits dans l'image où il se produit des changements brusque dans la fréquence de la couleur.

L'effet « sharpen » rend donc les contours plus prononcés.

La matrice de convolution peut être exprimée comme suit :

-1	-1	-1
-1	9	-1
-1	-1	-1

Ou encore comme suit :

0	-1	0
-1	5	-1
0	-1	0

La matrice de convolution précédente permet de ne tenir compte que des pixels voisins situés sur la verticale et l'horizontale éliminant ainsi 4 pixels de l'équation et par surcroît augmentant la vitesse de traitement.

Ce qui est important de retenir c'est que la somme des coefficients est toujours égale à 1. Peu importe le nombre de coefficients et leurs valeurs, la valeur centrale sera toujours positive.

Effet de relief (emboss)

Le filtre de relief produit une image dont les objets paraissent surélevés par rapport au fond de l'image. Il consiste à appliquer la matrice de convolution suivante :

-1	0	0
0	0	0
0	0	1

Ou encore comme suit :

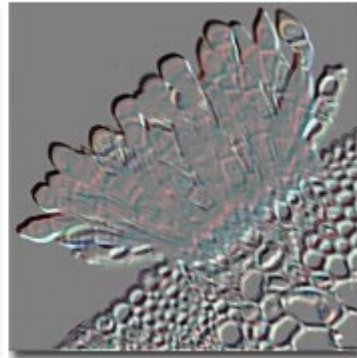
0	0	-1
0	0	0
1	0	0

Voici un exemple d'une image sur laquelle on a appliqué l'effet « emboss ».

Image originale



Image "emboss"



Effet aléatoire (random)

Ce filtre est assez simple à réaliser. Il consiste à prendre au hasard un pixel parmi les 8 voisins du pixel à traiter et de donner cette valeur au pixel à traiter.

Il n'y a pas de matrice de convolution disponible pour cet algorithme.

Détection des contours d'une image

La détection des contours est une opération très utile pour identifier les objets dans une image. Une fois que vous avez identifié les contours d'un objet, vous pouvez utiliser une méthode d'amélioration des contours pour faire en sorte que ces derniers aient une définition encore plus prononcées.

Détection de contour : méthode de « Sobel »

Cette technique utilise un simple filtre pour améliorer les contours des objets dans une direction définie. Par exemple, une variante de ce filtre permet de détecter les portions d'images dans lesquelles la couleur devient plus foncée en se déplaçant de droite à gauche dans l'image.

La détection des contours se réalise habituellement en utilisant un tableau de dimension finie qui contient une valeur pour chaque indice du tableau. Voici des exemples :

-1	-2	-1
0	0	0
1	2	1

Tableau 1 : Cette matrice permettra de détecter les contours horizontaux.

1	0	-1
2	0	-2
1	0	-1

Tableau 2 : Cette matrice permettra de détecter les contours verticaux.

La rangée ou la colonne de nombre positifs détermine le côté qui sera détecté.

Ainsi, dans le tableau 1, ce sont les contours situés en bas de l'objet qui sont détectés alors que le tableau 2 permet de détecter les contours situés à gauche des objets.

Méthode de calcul

La méthode est assez simple : vous multipliez tous les pixels par les valeurs correspondantes dans le tableau et vous faites l'addition. Ce résultat devient alors la nouvelle valeur du pixel du centre.

Exemple :

Soit la valeur des pixels suivant dans une portion d'une image :

50	50	50
50	128	128
50	128	128

Et le tableau de « Sobel » suivant pour détecter les contours situés à gauche:

1	2	1
0	0	0
-1	-2	-1

Le résultat du pixel central est :

$$50 * 1 + 50 * 2 + 50 * 1 + \\ 50 * 0 + 128 * 0 + 128 * 0 + \\ 50 * -1 + 128 * -2 + 128 * -1 = -234$$

qui devient 0 car on obtient une valeur < 0.

Caractéristiques du tableau de la méthode de « Sobel » :

1. On met les valeurs positives dans la colonne ou la rangée pour laquelle on veut détecter le contour. Par exemple, des valeurs positives dans la colonne de gauche permettra de détecter tous les contours situés à gauche des objets.
2. La somme de tous les coefficients donne toujours 0.

Détection de contour : méthode de « Laplace »

Cette méthode utilise un tableau ayant des valeurs symétriques pour détecter les contours dans toutes les directions. Ce type de détection est proche de la méthode de l'adoucissement à la différence que la somme de valeurs dans la méthode de Laplace donne 0. Ceci permet d'obtenir des valeurs de pixels qui tendent à devenir noir excepté aux endroits où on a détecté des hautes fréquences dans l'image. (Haute fréquence = contour)

Voici un exemple de tableau avec la méthode de Laplace

-1	-1	-1
-1	8	-1
-1	-1	-1

Conversion de couleurs à ton de gris

La conversion de couleurs à ton de gris est réalisée en faisant la moyenne des 3 composantes de la couleurs du pixel original. Cette méthode qui donne des résultats souvent éloignés de la réalité et n'est pas utilisée en pratique.

Ce qui est plutôt utilisée, c'est la formule de conversion NTSC qui est utilisée pour les images de télévision et de caméra en Amérique du Nord.

Cette formule est la suivante :

$$\text{Pixel_gris} = 0.299 * \text{Valeur_Pixel_Rouge} + 0.587 * \text{Valeur_Pixel_Vert} + 0.114 * \text{Valeur_Pixel_bleu}$$

Seuil de filtrage (Binarisation)

Les méthodes de seuil de filtrage permettent de convertir les images en ton de gris en image noir et blanc. C'est à dire comportant uniquement 2 tons de couleurs, soit le noir et le blanc. La technique consiste à faire passer un pixel à la valeur 0 ou 255 selon la valeur d'un seuil (threshold) prédéfini.

La première méthode est simple.

On sélectionne une valeur pour le seuil et applique les conditions suivantes :

SI $\text{Valeur_Pixel_Courant} < \text{Valeur_Seuil}$ alors $\text{Valeur_Pixel_Courant} = 0$

SI $\text{Valeur_Pixel_Courant} > \text{Valeur_Seuil}$ alors $\text{Valeur_Pixel_Courant} = 255$

Choix de la valeur du seuil

Seuil médian

La valeur du seuil peut être trouvée simplement en prenant la valeur médiane du spectre de couleur de 0 à 255. Ainsi, on obtient une valeur de seuil de 127 que l'on applique à l'image.

Cette méthode, bien que simple à exécuter, peut quand même donner des résultats intéressants mais s'éloigne souvent de la réalité. En effet, on prend pour acquis avec cette méthode que la moitié des pixels appartiennent aux objets et que l'autre moitié appartiennent au fond de l'image. Ceci est rarement le cas!!

Détermination du seuil à l'aide d'histogramme

Une autre méthode consiste à produire un histogramme de l'image considérée. Elle a pour avantage que le seuil est plus réaliste puisque la valeur correspond à la moyenne des valeurs de ton de gris rencontrées dans l'image. Mais qu'en est-il d'une image qui produit un histogramme qui comporte 2 maximum ?

Un histogramme consiste à garder dans un tableau le compte de la valeur de chaque pixel que l'on rencontre dans l'image.

Cette méthode n'en tient pas compte des cas où l'on rencontre plus d'un maximum dans l'image. Encore une fois, il est plutôt rare de voir des images qui comportent exactement 2 maximums bien définis.

Seuil de filtrage par sélection itérative

L'idée principale consiste à prendre la moyenne du niveau de gris du fond de l'image (S_b) et la moyenne du niveau de gris des objets (S_o) et de faire la moyenne de ces deux mesures ($S = (S_o + S_b) / 2$).

Voici la procédure :

1. On initialise les valeurs de S_o et S_b . Ces valeurs sont habituellement $S_o = 255$ et $S_b = 0$ puisqu'il est logique de croire que les objets seront blanc et le fond restera noir. On peut aussi donner la valeur médiane des valeurs de gris rencontrées dans l'image et initialiser la valeur S avec cette médiane. C'est ce qui est fait dans la suite de la procédure.
2. La prochaine étape consiste à raffiner les valeurs de S_b et S_o en utilisant la valeur du seuil S . En assumant que les objets plus foncés appartiennent au fond de l'image, S_o est recalculé en faisant la moyenne de tous les pixels ayant une valeur plus petite que la valeur de S . De la même façon, la valeur de S_b est calculée en faisant la moyenne de tous les pixels ayant une valeur plus grande que la valeur de S .

L'algorithme ressemble à ceci :

1. Prendre une première approximation du Seuil (S) en faisant la moyenne de l'intensité de tous les pixels de l'image.
2. SI Valeur_Pixel_Courant $\geq S$
 - Incrémente le compteur S_o . (Valeur des pixels appartenant à un objet)
 - Incrémente le compteur du nombre de pixel appartenant à un objet (N_o).SINON
 - Incrémente le compteur S_b (Valeur des pixels du fond de l'image).
 - Incrémente le compteur du nombre de pixel appartenant au fond (N_b).
3. $S = (S_o/N_o + S_b/N_b) / 2$
4. Revenir à l'étape 2 si la nouvelle valeur de S est égale à la valeur de S précédente. (Ou encore qu'il existe qu'une différence de 1%)

Analyse d'image

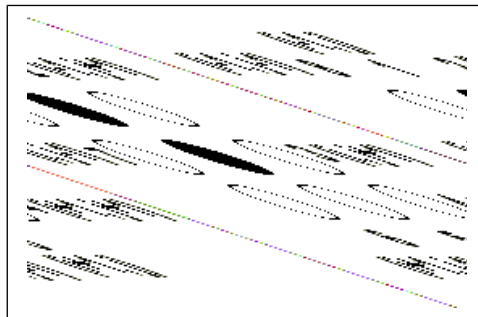
En général, la première étape dans le traitement d'une image bi-niveaux est de distinguer les objets de leurs fonds. Une fois cette étape terminée, on peut tenter de reconnaître certains objets en récupérant des caractéristiques comme :

- la forme
- la grandeur
- l'orientation

et en comparant ces mesures à celles d'objets connus du même type.

Identification des régions d'une image

On doit identifier les pixels qui sont connectés entre eux. C'est à dire que si on prend un pixel et qu'on regarde ses voisins, il doit y avoir au moins un d'entre eux qui soit allumés pour que ces pixels soient **connectés**. La figure suivante illustre mieux les propos précédents.



Dans la figure précédente, le pixel numéroté 1 et le pixel du centre sont connectés. L'algorithme d'identification d'une région devient ainsi relativement simple :

1. Il s'agit de trouver le premier pixel de niveau noir ou blanc.
2. Marquer la région contenant ce pixel.

Si on décortique le point #2 de l'algorithme, on obtient :

```
POUR Rangée de -1 à 1 FAIRE
  POUR Colonne de -1 à 1 FAIRE
    SI (Colonne = 0) ET (Rangée = 0) ALORS retourner au début de
    la boucle interne
    SI (Rangée * Colonne <> 0) ALORS retourner au début de la
    boucle interne

    SI (Pixel[i+Rangée][j+Colonne] = 0) ALORS Pixel[i+Rangée]
    [j+Colonne] = 1
  FIN POUR
FIN POUR
```

Ou encore de façon récursive :

1. Recherche d'un pixel source (Seed pixel)
2. Étiquetage des pixels trouvés.
 1. Recherche d'un pixel source

```
Etiquete = 1
POUR y = 0 à RésolutionY - 1
  POUR x = 0 à RésolutionX - 1
    SI Pixel = 0
      Analyser_Region(Etiquette, x, y)
      Incréments Étiquette
    FINPOUR
  FINPOUR
FINPOUR
```

2. Étiquetage des pixels trouvés. (Analyser_Region(Etiquette, x, y))

```

SI Pixel = 0 //Noir
  Pixel = Etiquette
// On analyse les quatres directions

  SI x < ResolutionX - 1
    ?? Analyser_Region(Etiquette, x+1,y)

  SI x > 0
    ?? Analyser_Region(Etiquette, x-1,y)

  SI y < ResolutionY - 1
    ?? Analyser_Region(Etiquette, x,y+1)

  SI y > 0
    ?? Analyser_Region(Etiquette, x,y-1)

```

Exemple :

Soit l'image des pixels suivants (image 4 x 4 !! pour simplifier)

25	255	0	0
5			
0	0	0	0
0	0	255	255
0	0	255	255

```

Etiquette = 1
Analyser_Region(1,0,0)           Etiquette = 1, x = 0, y = 0
|
|   Analyser_Region(1,1,0)
|   |
|   |   Analyser_Region(1,2,0) // Pixel à 0
|   |   Analyser_Region(1,0,0) // Déjà à 1
|   |   Analyser_Region(1,1,1) // Pixel à 0
|   |
|   Analyser_Region(1,0,1) // Pixel à 0
|

```

On cherche ensuite la prochaine source de gauche à droite et de haut en bas.

À partir de cette source, on analyse la région dans l'ordre suivant : droite, gauche, haut, bas.

Quand on ne peut aller plus loin, on revient à la case précédente et on continue les directions.

Etiquette = 2

Analyser_Region(2,2,2) //Pixel à 255 devient 2

 Analyse_Region(2,3,2) // Pixel à 255 devient 2
 Analyser_Region(2,2,2) // Pixel déjà à 2
 Analyser_Region(2,2,1) // Pixel à 0

Analyser_Region(2,2,3) // Pixel à 255 devient 2

Propriétés mesurables des régions

Connaissant certaines propriétés des images bi-niveaux ainsi que certaines définitions inhérentes à ces images, nous pouvons procéder à certains calculs. Certains systèmes de vision ne s'attardent qu'à mesurer les dimensions d'un objet dans une image car ce sont ces mesures qui permettent de trouver la position des objets. Dans le cas des images bi-niveaux, ce sont la forme, la position et l'orientation qui donne l'information voulue. Ces informations deviennent donc cruciales et permettent même de décrire la forme d'objets complexes en combinant des objets de formes plus simples.

Voyons donc quelques une de ces propriétés.

Aire

Principe :

L'aire d'une région est exprimée par le nombre de pixels appartenant à une région déterminée. L'aire physique est obtenue en multipliant le nombre de pixels par l'aire réelle de ce pixel. Par exemple, une région contenant 10 pixels, chacun correspondant à 1.2 cm^2 aura une aire physique réelle de $10 \times 1.2 = 12 \text{ cm}^2$. Cependant, il arrive souvent que l'aire est exprimée en nombre de pixel seulement.

Algorithme :

1. Une région est marquée d'une valeur avec l'aide de l'algorithme discuté précédemment.
2. Le nombre de pixel ayant cette valeur est ensuite compté et donne l'aire.

Périmètre

Principe:

Il s'agit de trouver l'ensemble des pixels qui font partie de l'objet et qui ont au moins un voisin qui appartient au fond (background).

Il faut cependant déterminer si les pixels sont connectés sous une forme de 4-adjacence ou de 8-adjacence.

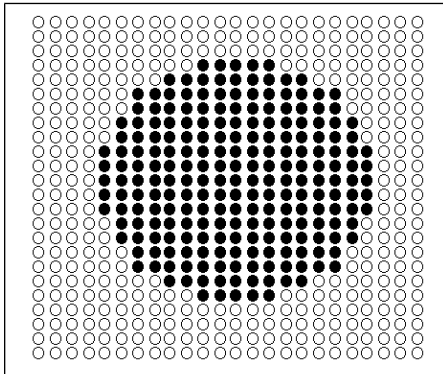
Algorithme:

1. Identifier les pixels sur la région.
2. Chaque pixel situé sur le périmètre et ses voisins sont comparés à un "template" pour connaître la distance (poids) qui doit être assignée à ce pixel.
3. Faire la somme de toutes les distances (poids) trouvées précédemment. Ceci donne le périmètre approximé.

Technique utilisée pour trouver la distance (poid) associée à un pixel

L'algorithme calculant le périmètre d'un objet, de même que plusieurs autres algorithmes que nous verrons plus loin, utilise une technique qui permet d'associer un poids à un pixel. Ce poids est donné en relation avec la position du pixel analysé et de ses pixels voisins. Utilisons un exemple pour fixer les idées.

Soit le cercle suivant:

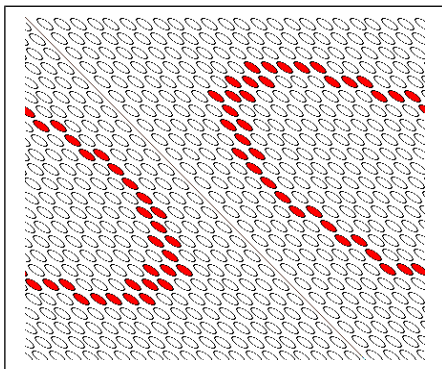


L'image ci-contre représente un objet circulaire ayant un diamètre de 17 pixels. Le périmètre mathématique de cet objet devrait donner:

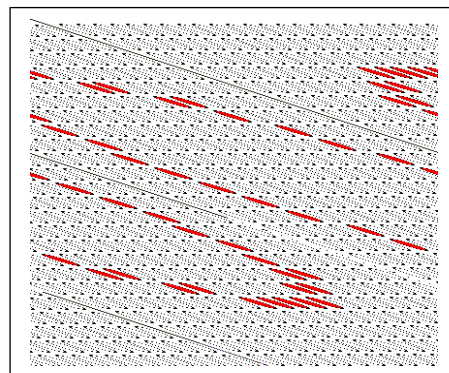
où r : le rayon du cercle.

Or le diamètre étant de 17 pixels, on obtient:

Vous savez que les pixels peuvent être connectés sous une forme 4-adjacente ou 8-adjacente. Les deux figures suivantes montrent le résultat du périmètre précédent en utilisant la 4-adjacence et la 8-adjacence:



4 adjacence
Périmètre = 64



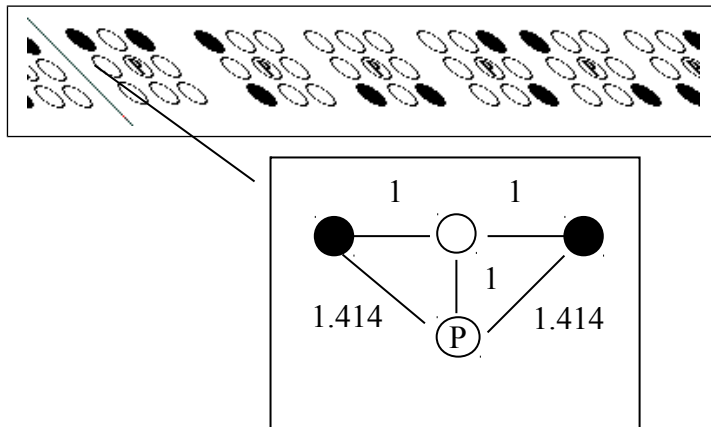
8 adjacence
Périmètre = 48

On voit clairement qu'il y a une grande erreur de propagation par rapport à la valeur réelle mathématique. En plus, les valeurs diffèrent entre la 8 et la 4 adjacence!! La raison en est fort simple, c'est qu'un pixel représente non pas une distance linéaire mais plutôt une surface. Une hypothèse de base que l'on peut accepter concernant les pixels, c'est que ceux-ci représentent une région rectangulaire et il y a, en conséquence, plusieurs façons qu'un périmètre d'une figure puisse passer au travers d'un pixel.

Nous sommes tentés de compter le nombre de pixel tout simplement et c'est là l'erreur. La solution consiste à donner un poids au pixel considéré selon le "pattern" observé avec ses voisins. Ainsi, dépendamment du "pattern", un pixel aura tantôt la valeur 1, tantôt la valeur 0.5, etc.

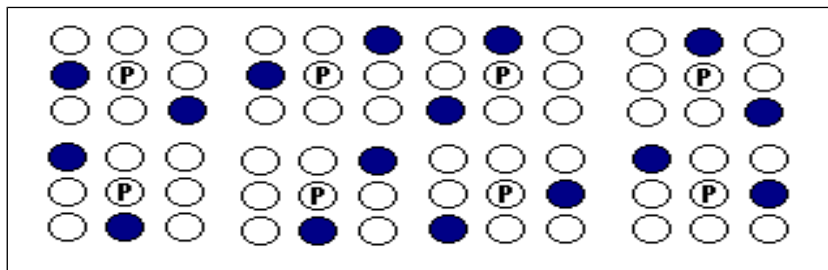
C'est donc plus logique d'essayer de trouver quelle est la distance représentée par un pixel et de donner un poids à ce dernier. Il en résulte les cas suivants:

- Pixel ayant des voisins en diagonale



Les pixels marqués P ont une "longueur" effective de 1.414. En effet, comme le montre la figure ci-contre, le pixel P obtient un poids qui représente une distance de 1 unité par 1 unité.

- Pixel ayant des voisins en diagonale et sur la verticale ou l'horizontale



Le pixel P obtient un poids équivalent à $\frac{1}{2}$ fois la longueur de 1 pixel d'unité plus $\frac{1}{2}$ fois la longueur d'une diagonale de 1.414 unités. Ainsi, le pixel P obtient un poids de:

$$P_{\text{poids}} = \frac{1}{2} + 1.414/2 = 0.5 + 0.707 = 1.207$$

Reprenons l'exemple du cercle ayant un diamètre de 17 pixels. En utilisant la 8 adjacence et en donnant des poids au pixel selon les techniques précédentes, on obtient un périmètre de 54.6 dont ce dernier est à 2% prêt de la valeur réelle.

Centre de masse

Le centre de masse donne la valeur (x,y) d'un pixel d'un objet pour lequel il y a un nombre égaux de pixels qui entoure ce dernier.

Dans le cas d'un crayon, c'est le point pour lequel le crayon sera en équilibre. Par conséquence, le centre de masse est souvent appelé **Centre de gravité**.

Le centre de masse peut être définie avec les équations suivantes:

$$C_r = \frac{\sum_{rangée=1}^{NR} \sum_{Col=1}^{NC} rangée * 1}{Aire}$$

$$C_c = \frac{\sum_{rangée=1}^{NR} \sum_{Col=1}^{NC} Col * 1}{Aire}$$

Circularité

Cette méthode permet de trouver le degré de **circularité** d'une pièce. En fait, elle calcule le rapport de l'aire de la pièce par rapport au carré de son périmètre. Dans le cas d'un cercle parfait, ce rapport donne une valeur égale à 1.

Soit l'aire du cercle et le périmètre de ce dernier:

$$A = \pi R^2$$

$$P = 2\pi R = \pi D$$

L'équation suivante peut-être utilisée pour le calcul de la circularité:

$$\text{Circularité} = \frac{P^2}{4 * \pi * A}$$

Voici quelques exemples de valeur du calcul de circularité pour des figures géométriques usuelles:

- Cercle parfait : 1.0
- Cercle dessiné avec un logiciel : 1.02 à 1.1
- Rectangle (carré) : 1.206
- Triangle : 3.308
- Polygone : 1.771

Rectangularité

Cette mesure permet d'obtenir une approximation de la forme rectangulaire d'un objet. On peut donc savoir si un objet est de type rectangulaire ou s'il s'en éloigne.

La rectangularité est définie comme le rapport de l'aire de l'objet sur l'aire du plus petit rectangle qui englobe l'objet. Mathématiquement :

$$R = A_{\text{objet}} / A_{\text{min}}$$

Où : A_{objet} est l'aire de l'objet

A_{min} peut être calculée en prenant les coordonnées du pixel le plus en haut et à gauche dans l'objet (ceci devient le x minimum et le y minimum) et les coordonnées du pixel le plus en bas et à droite dans l'objet (ceci devient le x maximum et le y maximum).

On obtient donc les coordonnées suivantes :

$(X_{\text{min}}, Y_{\text{min}})$ et $(X_{\text{max}}, Y_{\text{max}})$

et l'aire A_{min} est donnée par :

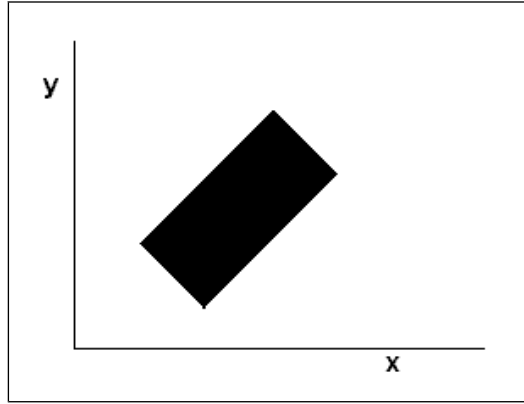
$$A_{\text{min}} = (X_{\text{max}} - X_{\text{min}}) (Y_{\text{max}} - Y_{\text{min}})$$

La valeur du calcul de rectangularité devrait donner 1 pour un rectangle ou un carré. La valeur est entre 0 et 1 pour une forme qui s'éloigne d'une forme rectangulaire.

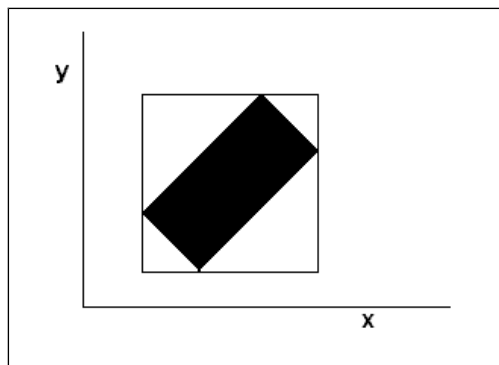
La procédure précédente fonctionne très bien pour des rectangles ou des carrés qui ont la base parallèle à l'axe des x de l'image.

Qu'en est-il d'une figure rectangulaire qui ferait un angle par rapport à l'axe des x de l'image ?

Il en résulte une rectangularité qui s'éloigne de la valeur attendu qui est proche de 1. La figure suivante illustre les propos :



Si on applique l'algorithme discuté plus haut, on obtient :



On obtient une valeur de rectangularité qui s'approche de 0.4 alors que celle-ci devrait nous donner une valeur proche de 1.