

Analyse d'images : Comparaison des performances de détecteurs de contours

1. Objectif

Le module Atelier2 est destiné à répondre à un problème de physique appliquée / sciences de l'ingénieur donné (un même problème avec plusieurs solutions), par l'étude et le choix de solutions théoriques (par les élèves répartis en équipes) et réaliser la simulation / interprétation / comparaison / visualisation des résultats.

2. Sujet

L'extraction de contours en analyse d'images est une méthode de segmentation permettant d'isoler les différentes régions d'une image, autorisant des traitements conséquents de plus haut niveau (reconnaissance de forme ...)

Le problème du choix d'un détecteur de contours pour la segmentation d'images se pose du fait de la pluralité d'algorithmes d'extraction de contours : opérateurs G4dif (Gradient à 4 différences), Gnabla (dérivation discrète par différences finies), détecteurs de Roberts, de Prewitt, de Sobel, de Kirsch, de Robison ... (voir *Documentation plus loin*)

Dans le but de fournir une aide à la décision en matière de choix d'un détecteur de contours, le thème consiste à produire des indicateurs mathématiques destinés à comparer les détecteurs de contours appliqués sur des mêmes images de luminance, permettant de classer les détecteurs à partir de leurs performances (**en incluant d'autres détecteurs de contours que les précédents**).

A partir d'images tests de luminance (images de synthèse en niveaux de gris, par exemple une mire avec une transition unique verticale, horizontale ... entre deux régions uniformes avec un contraste donné - voir *Documentation plus loin*) bruitées artificiellement (bruit de puissance donnée p avec $0 \leq p \leq 15000$ - voir *Documentation plus loin*), plusieurs critères de performances peuvent être envisagés : (**en ajoutant d'autres critères indicateurs de performances**), par exemple :

- **Q** : la *qualité* de la détection (points de contours détectés, manqués, fausses détections)
- **P** : la *précision* de localisation des points de contours détectés
- **F** : la *finesse* des contours détectés

...

Le seuil de détection de contours étant un paramètre de réglage influant sur les indicateurs et propre à chaque méthode de détection (un seuil identique pour tous les détecteurs ne serait pas cohérent), chaque indicateur peut être calculé pour toutes les valeurs de seuil pour ne retenir *in fine* comme résultat final d'indicateur uniquement le meilleur score.

Chaque classe, divisée en groupes d'environ 4 élèves - maximum - traitera de ce même thème. Chaque groupe apportera sa solution personnelle (détecteurs de contours à comparer / classer, réglages des paramètres, indicateurs de performances, interprétation et visualisation des résultats) à ce même problème. Une gestion de projet (découpage / répartition des tâches aux membres de groupe / planification) est vivement recommandée.

3. Outil de simulation (~3 semaines)

Les langages de simulation mathématique **Mathcad** (**conseillé**) ou **Scilab** sont proposés.

Prise en main du langage de simulation appliqué au traitement d'images (programmes de traitements basiques : Négatif d'une image, Binarisation, Gradient ...) Programmation des détecteurs de contours.

4. Etude théorique (~5 semaines)

Etude bibliographique. Définition des critères et indicateurs de performances destinés à comparer et classer les détecteurs de contours en fonction de leurs performances. Test et validation des critères.

5. Simulation (~6 semaines) + Présentation/Soutenance (1 semaine)

Simulation avec l'outil de programmation :

- Implémentation des indicateurs de performances
- Réglages des paramètres (puissance de bruit , seuil ...)
- Calcul des résultats des indicateurs
- Tests - Présentation - Visualisation / Interprétation des résultats de classification des indicateurs
- Rapport

Un rapport et une soutenance de présentation du travail réalisé (en dernière séance) par groupe (avec questions de l'enseignant) de la *totalité* du projet « Etude théorique + Simulation » sont à effectuer. Fournir une archive zip contenant rapport pdf, sources **Mathcad**, images, documentation personnelle, bibliographie ... ainsi nommée :

P2Cx_noms_élèves (par ordre alphabétique).zip où x est la classe (1, 2 ...).

La note tiendra compte notamment des points suivants :

- Pertinence / Nombre des indicateurs de performance
- Nombre de détecteurs de contours comparés
- Visualisation des résultats de classification des indicateurs
- Aide à la décision apportée quant au choix d'un détecteur de contours
- Qualité du rapport

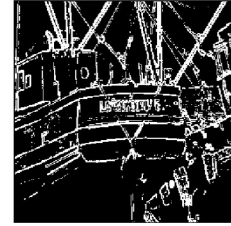
6. Exemples de résultats de détecteurs de contours



I image de luminance initiale



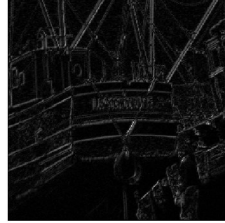
J = G4dif(I) norme de gradient



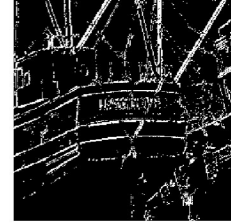
contours K = Binarisation(J, seuil) seuil = 50



I image de luminance initiale



J = Gnabla(I) norme de gradient



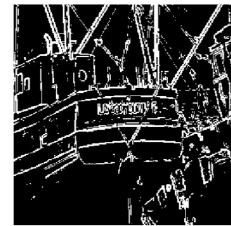
contours K = Binarisation(J, seuil) seuil = 40



I image de luminance initiale



J = Roberts(I) norme de gradient



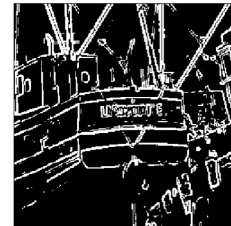
contours K = Binarisation(J, seuil) seuil = 50



I image de luminance initiale



J = Prewitt(I) norme de gradient



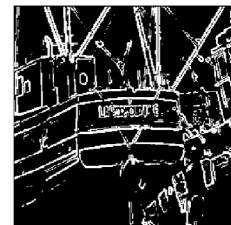
contours K = Binarisation(J, seuil) seuil = 55



I image de luminance initiale



J = Sobel(I) norme de gradient



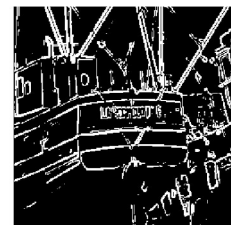
contours K = Binarisation(J, seuil) seuil = 55



I image de luminance initiale



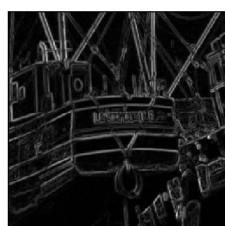
J = Kirsch(I) v1 norme de gradient



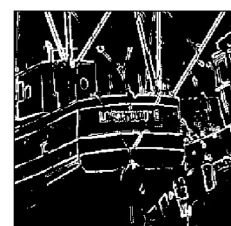
contours K = Binarisation(J, seuil) seuil = 55



I image de luminance initiale



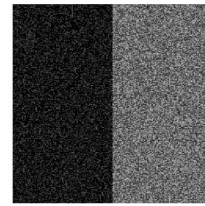
J = Robinson(I) norme de gradient



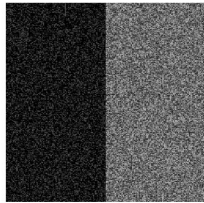
contours K = Binarisation(J, seuil) seuil = 55



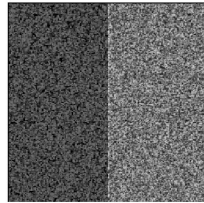
U mire size 256 mini 10 contrast 100 X = 128



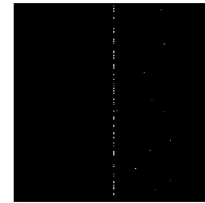
I = Bruitage(U, p) mire bruitée p = 2000



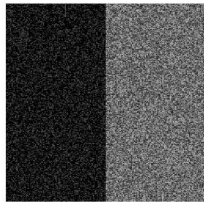
I mire bruitée p = 2000



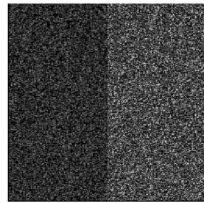
J = G4dif(I) norme de gradient



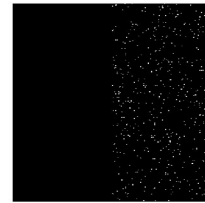
contours **K = Binarisation(J, seuil)** seuil = 210



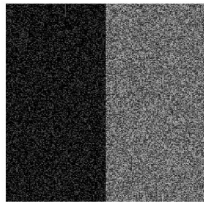
I mire bruitée p = 2000



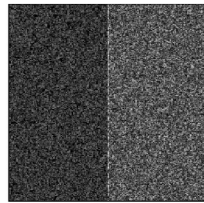
J = Gnabla(I) norme de gradient



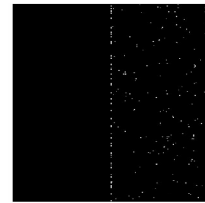
contours **K = Binarisation(J, seuil)** seuil = 200



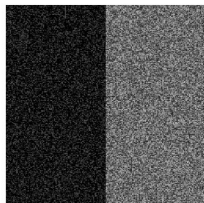
I mire bruitée p = 2000



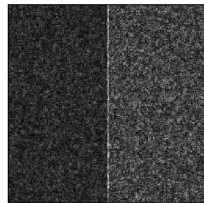
J = Roberts(I) norme de gradient



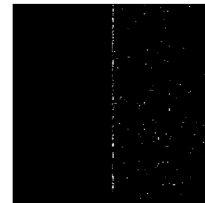
contours **K = Binarisation(J, seuil)** seuil = 190



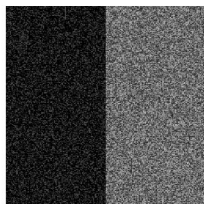
I mire bruitée p = 2000



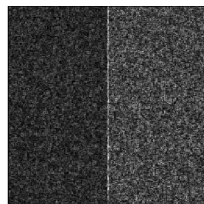
J = Prewitt(I) norme de gradient



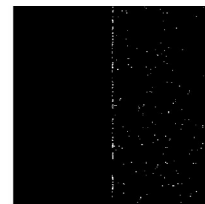
contours **K = Binarisation(J, seuil)** seuil = 170



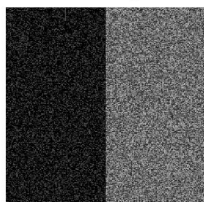
I mire bruitée p = 2000



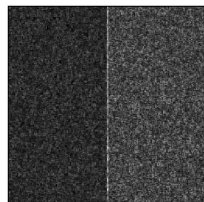
J = Sobel(I) norme de gradient



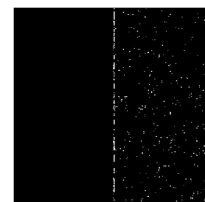
contours **K = Binarisation(J, seuil)** seuil = 170



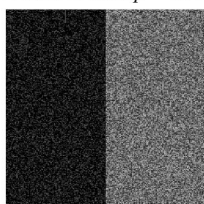
I mire bruitée p = 2000



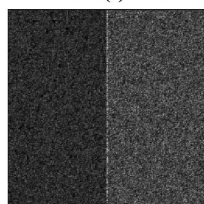
J = Kirsch(I) v1 norme de gradient



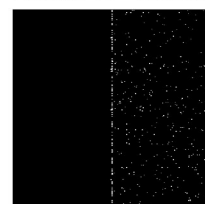
contours **K = Binarisation(J, seuil)** seuil = 150



I mire bruitée p = 2000



J = Robinson(I) norme de gradient



contours **K = Binarisation(J, seuil)** seuil = 150

7. Documentation

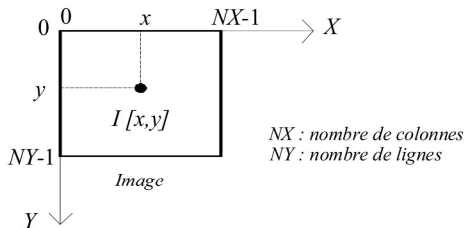
Introduction : Mémoire d'image (frame buffer)

Image numérique de luminance (Noir & Blanc) *bitmap* ou *raster* = tableau 2D I de NY lignes et NX colonnes (NX, NY entiers > 0)

$I[x, y]$ = niveau de gris du pixel de coordonnées x et y : $0 \leq x \leq NX-1$ $0 \leq y \leq NY-1$ (x, y entiers)

$I[x, y]$ encore noté $I_{x,y}$ est représenté sous *Mathcad* par l'élément de tableau $I_{y,x}$ (convention ligne-colonne de la notation matricielle)

$I[x, y]$ entier codé sur 8 bits (256 nuances de gris) : 0 (noir) $\leq I[x, y] \leq 255$ (blanc)



L'image peut aussi être mémorisée sous forme 1D $I[i]$ avec le calcul de l'index i du pixel (x, y) : $i = y \cdot NX + x$.

Codage RVB d'une image numérique couleur : la couleur d'un pixel est codée sur 3 octets (24 bits) avec 1 octet par couleur R (Rouge), V (Vert) ou B (Bleu) ($2^{24} \approx 16.7$ millions de couleurs).

Binarisation

Chaque pixel, de niveau de gris n représenté par un entier compris entre 0 (noir) et 255 (blanc) (codage sur 8 bits) est ramené à :

- . blanc (niveau de gris 255) si le niveau de gris n est supérieur au seuil
- . noir (niveau de gris 0) sinon.

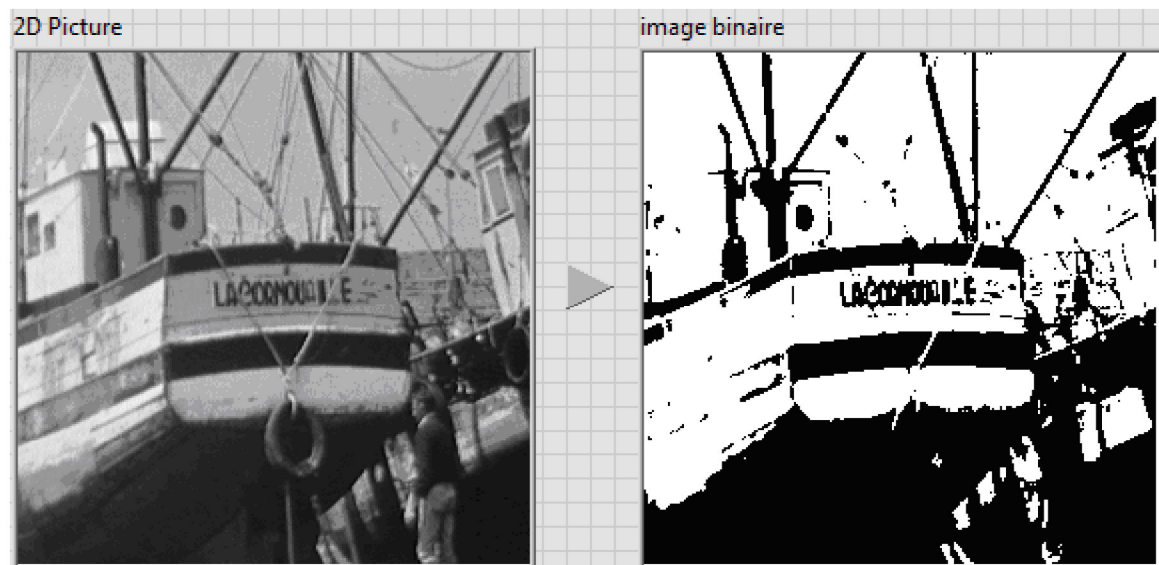
l'image traitée est stockée dans un tableau différent de celui de l'image initiale.

Fonction Binarisation(I, seuil)

Algorithme de binarisation de l'image I pour produire l'image binaire J (le paramètre *seuil* est un entier compris entre 0 et 255) :

$$\text{pour tout pixel de coordonnées } x \text{ et } y \text{ (indices de colonne et de ligne) de l'image de luminance } I : \begin{cases} J_{x,y} = 255 & \text{si } I_{x,y} \geq \text{seuil} \\ J_{x,y} = 0 & \text{si } I_{x,y} < \text{seuil} \end{cases}$$

Exemple de binarisation :



I image initiale de luminance

$J = \text{Binarisation}(I, \text{seuil})$ image binarisée *seuil* = 110

La détection de contours

▪ **Approche gradient** : à partir de l'image I de luminance (image en niveaux de gris) l'image du gradient est calculée. Les points de contours sont ensuite obtenus par binarisation de l'image G du module du gradient : les points de l'image du module du gradient de valeur supérieure à un seuil sont élus comme points de contours :

- calcul de l'image G du module du gradient à partir de I
- binarisation de G pour obtenir l'image de contours C : $C = \text{Binarisation}(G, \text{seuil})$

▪ **Approche laplacien** : les passages par zéro du laplacien (gradient du gradient) sont utilisés.

L'image du module du gradient peut être obtenue :

- comme un maximum de différences (opérateur **G4dif**)
- par filtrage linéaire (opérateurs **Gnabla, Roberts, Prewitt, Sobel, Kirsch, Robinson ...**) : l'image G du module du gradient est alors obtenue par convolution (symbolisée par l'opérateur $*$) de l'image I de luminance avec un - ou plusieurs - filtre « dérivateur » de masque M : $G = I * M$

Algorithme de convolution de l'image I avec un filtre de masque M de taille 3x3 pixels pour produire l'image G :

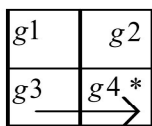
pour tout pixel de coordonnées x et y (indices de colonne et de ligne) de l'image I :

$$G_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot M_{i,j}$$

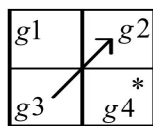
▪ **Opérateur G4dif de calcul du module du gradient G**

Algorithme du calcul de l'image G module du gradient de l'image I en niveaux de gris :

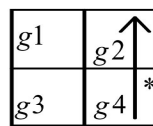
pour tout pixel de coordonnées x et y (indices de colonne et de ligne) (noté $*$) de l'image de luminance I , 4 différences sont calculées $\Delta_i[x, y]$ ($i=0$ à 3), via un masque de taille 2x2 pixels :



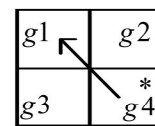
$$\Delta_0[x, y] = g4 - g3$$



$$\Delta_1[x, y] = g2 - g3$$



$$\Delta_2[x, y] = g2 - g4$$



$$\Delta_3[x, y] = g1 - g4$$

(* : pixel central d'application du masque (x,y)
 g_i ($i=0$ à 3) : niveau de gris d'un pixel

La norme du gradient $G_{x,y}$ au point (x, y) est obtenue par : $G_{x,y} = \text{Sup}_{\text{pour } i=0 \rightarrow 3} |\Delta_i[x, y]|$

plus grande différence de niveaux de gris en valeur absolue dans le voisinage 2x2, plus grande étendue

▪ **Filtre Gnabla de calcul du module du gradient G**

Algorithme du calcul de l'image G module du gradient de l'image de luminance I en niveaux de gris :

pour tout pixel de coordonnées x et y (indices de colonne et de ligne) de l'image I , le masque de taille 3x3 pixels du filtre

$$M = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 2 & 0 \\ 0 & 0 & 0 \end{pmatrix} \text{ est appliqué pour produire le gradient au point } (x, y) \text{ par convolution : } g_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot M_{i,j}$$

L'image g peut être obtenue directement par la fonction intégrée *Mathcad* : $g = \text{convol2d}(I, \mathbf{M})$ avec $\mathbf{M} = \begin{pmatrix} 0 & -1 \\ -1 & 2 \end{pmatrix}$

La norme du gradient $G_{x,y}$ au point (x, y) est obtenue par : $G_{x,y} = |g_{x,y}|$

▪ **Filtre de Roberts de calcul du module du gradient G**

Algorithme du calcul de l'image G module du gradient de l'image de luminance I en niveaux de gris :

pour tout pixel de coordonnées x et y (indices de colonne et de ligne) de l'image I , les masques

$$\left\{ \begin{array}{l} H1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \\ H2 = \begin{pmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \end{array} \right. \text{ sont}$$

appliqués pour produire les 2 gradients directionnels au point (x, y) par convolution :

$$\left\{ \begin{array}{l} g1_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot H1_{i,j} \\ g2_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot H2_{i,j} \end{array} \right.$$

Les images $g1$ et $g2$ peuvent être obtenues directement par la fonction intégrée *Mathcad* :

$$\left\{ \begin{array}{l} g1 = \text{convol2d}(I, \mathbf{H1}) \\ g2 = \text{convol2d}(I, \mathbf{H2}) \end{array} \right.$$

avec

$$\left\{ \begin{array}{l} \mathbf{H1} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \\ \mathbf{H2} = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \end{array} \right.$$

La norme du gradient $G_{x,y}$ au point (x, y) est obtenue par : $G_{x,y} = \sqrt{g1_{x,y}^2 + g2_{x,y}^2}$

▪ **Filtre de Prewitt de calcul du module du gradient G**

Algorithme du calcul de l'image G module du gradient de l'image de luminance I en niveaux de gris :

pour tout pixel de coordonnées x et y (indices de colonne et de ligne) de l'image I , les masques

$$\left\{ \begin{array}{l} H1 = \frac{1}{3} \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} \\ H2 = \frac{1}{3} \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \end{array} \right. \text{ sont}$$

appliqués pour produire les 2 gradients directionnels au point (x, y) par convolution :

$$\left\{ \begin{array}{l} g1_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot H1_{i,j} \\ g2_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot H2_{i,j} \end{array} \right.$$

Les images $g1$ et $g2$ peuvent être obtenues directement par la fonction intégrée *Mathcad* :

$$\left\{ \begin{array}{l} g1 = \text{convol2d}(I, H1) \\ g2 = \text{convol2d}(I, H2) \end{array} \right.$$

La norme du gradient $G_{x,y}$ au point (x, y) est obtenue par : $G_{x,y} = \sqrt{g1_{x,y}^2 + g2_{x,y}^2}$

▪ **Filtre de Sobel de calcul du module du gradient G**

Algorithme du calcul de l'image G module du gradient de l'image de luminance I en niveaux de gris :

pour tout pixel de coordonnées x et y (indices de colonne et de ligne) de l'image I, les masques
$$\left\{ \begin{array}{l} H1 = \frac{1}{4} \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \\ H2 = \frac{1}{4} \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \end{array} \right. \text{ sont}$$

appliqués pour produire les 2 gradients directionnels au point (x, y) par convolution :
$$\left\{ \begin{array}{l} g1_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot H1_{i,j} \\ g2_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot H2_{i,j} \end{array} \right.$$

Les images g1 et g2 peuvent être obtenues directement par la fonction intégrée *Mathcad* :
$$\left\{ \begin{array}{l} g1 = \text{convol2d}(I, H1) \\ g2 = \text{convol2d}(I, H2) \end{array} \right.$$

La norme du gradient $G_{x,y}$ au point (x, y) est obtenue par : $G_{x,y} = \sqrt{g1_{x,y}^2 + g2_{x,y}^2}$

▪ **Filtre de Kirsch (version 1) de calcul du module du gradient G**

Algorithme du calcul de l'image G module du gradient de l'image de luminance I en niveaux de gris :

pour tout pixel de coordonnées x et y (indices de colonne et de ligne) de l'image I, les masques
$$\left\{ \begin{array}{l} H1 = \frac{1}{15} \begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix} \\ H2 = \frac{1}{15} \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix} \\ H3 = \frac{1}{15} \begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix} \\ H4 = \frac{1}{15} \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \end{array} \right. \text{ sont}$$

appliqués pour produire les 4 gradients directionnels au point (x, y) par convolution :
$$\left\{ \begin{array}{l} g1_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot H1_{i,j} \\ g2_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot H2_{i,j} \\ g3_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot H3_{i,j} \\ g4_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot H4_{i,j} \end{array} \right.$$

Les images g1 ... g4 peuvent être obtenues directement par la fonction intégrée *Mathcad* :
$$\left\{ \begin{array}{l} g1 = \text{convol2d}(I, H1) \\ g2 = \text{convol2d}(I, H2) \\ g3 = \text{convol2d}(I, H3) \\ g4 = \text{convol2d}(I, H4) \end{array} \right.$$

La norme du gradient $G_{x,y}$ au point (x, y) est : $G_{x,y} = \left| g_{x,y} \right|$ avec $g_{x,y} = \left[g1_{x,y} \quad g2_{x,y} \quad g3_{x,y} \quad g4_{x,y} \right]^T$

▪ **Filtre de Kirsch (version 2) de calcul du module du gradient G**

Algorithme du calcul de l'image G module du gradient de l'image de luminance I en niveaux de gris :

pour tout pixel de coordonnées x et y (indices de colonne et de ligne) de l'image I , les masques

$$\left. \begin{aligned} H0 &= \frac{1}{15} \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \\ H1 &= \frac{1}{15} \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix} \\ H2 &= \frac{1}{15} \begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix} \\ H3 &= \frac{1}{15} \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{pmatrix} \\ H4 &= \frac{1}{15} \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix} \\ H5 &= \frac{1}{15} \begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix} \\ H6 &= \frac{1}{15} \begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix} \\ H7 &= \frac{1}{15} \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix} \end{aligned} \right\} \text{ sont}$$

appliqués pour produire les 8 gradients directionnels au point (x, y) par convolution :

$$\left\{ \begin{aligned} gi_{x,y} &= \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot Hi_{i,j} \\ i &= 0 \rightarrow 7 \end{aligned} \right.$$

Les images $g0 \dots g7$ peuvent être obtenues directement par la fonction intégrée *Mathcad* :

$$\left\{ \begin{aligned} gi &= \text{convol2d}(I, Hi) \\ i &= 0 \rightarrow 7 \end{aligned} \right.$$

La norme du gradient $G_{x,y}$ au point (x, y) est obtenue par :

$$G_{x,y} = \underset{\text{pour } i=0 \rightarrow 7}{\text{Sup}} \left| gi_{x,y} \right|$$

▪ **Filtre de Robison de calcul du module du gradient G**

Algorithme du calcul de l'image G module du gradient de l'image de luminance I en niveaux de gris :

pour tout pixel de coordonnées x et y (indices de colonne et de ligne) de l'image I , les masques

$$\left\{ \begin{array}{l} H1 = \frac{1}{5} \begin{pmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{pmatrix} \\ H2 = \frac{1}{5} \begin{pmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \\ H3 = \frac{1}{5} \begin{pmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & -1 & 1 \end{pmatrix} \\ H4 = \frac{1}{5} \begin{pmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{pmatrix} \end{array} \right. \text{sont}$$

appliqués pour produire les 4 gradients directionnels au point (x, y) par convolution :

$$\left\{ \begin{array}{l} g1_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot H1_{i,j} \\ g2_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot H2_{i,j} \\ g3_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot H3_{i,j} \\ g4_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-j} \cdot H4_{i,j} \end{array} \right.$$

Les images $g1 \dots g4$ peuvent être obtenues directement par la fonction intégrée *Mathcad* :

$$\left\{ \begin{array}{l} g1 = \text{convol2d}(I, H1) \\ g2 = \text{convol2d}(I, H2) \\ g3 = \text{convol2d}(I, H3) \\ g4 = \text{convol2d}(I, H4) \end{array} \right.$$

La norme du gradient $G_{x,y}$ au point (x, y) est : $G_{x,y} = |g_{x,y}|$ avec $g_{x,y} = [g1_{x,y} \quad g2_{x,y} \quad g3_{x,y} \quad g4_{x,y}]^T$

Fonction $bbc(N, p)$ générant un bruit 1D de N points, de puissance p à partir de la fonction intégrée Mathcad $whiten(N)$:

$$bbc(N, p) \equiv \sqrt{12 \cdot p} \cdot whiten(N)$$

Fonction $bruitage(I, p)$ de bruitage d'une image I (avec une puissance de bruit p) à partir de la fonction $bbc(N, p)$:

$$\text{Bruitage}(I, p) \equiv \left| \begin{array}{l} N \leftarrow \text{rows}(I) \cdot \text{cols}(I) \\ b \leftarrow \text{bbc}(N, p) \\ n \leftarrow 0 \\ \text{for } y \in 0.. \text{rows}(I) - 1 \\ \quad \text{for } x \in 0.. \text{cols}(I) - 1 \\ \quad \quad \left| \begin{array}{l} J_{y,x} \leftarrow \text{trunc}(I_{y,x} + b_n) \\ n \leftarrow n + 1 \end{array} \right. \\ \quad \text{for } y \in 0.. \text{rows}(J) - 1 \\ \quad \quad \text{for } x \in 0.. \text{cols}(J) - 1 \\ \quad \quad \quad \left| \begin{array}{l} J_{y,x} \leftarrow 255 \text{ if } J_{y,x} > 255 \\ J_{y,x} \leftarrow 0 \text{ if } J_{y,x} < 0 \end{array} \right. \end{array} \right. J$$

Image mire de taille $size$, avec une transition verticale localisée en X , de contraste $contrast$ à partir d'une luminance $mini$:

$$\text{mire}(size, mini, contrast, X) \equiv \left| \begin{array}{l} \text{for } y \in 0.. size - 1 \\ \quad \text{for } x \in 0.. size - 1 \\ \quad \quad \left| \begin{array}{l} I_{y,x} \leftarrow mini \text{ if } x < X \\ I_{y,x} \leftarrow mini + contrast \text{ otherwise} \end{array} \right. \end{array} \right. I$$