



BLIN Sabine, LARCHER Elya, MONIN Louis,
SASTRE-GARAU Pablo, VINCENT Erine, LAMRINI Mohamed-Amin
ING2 MI2 – Mathématiques Appliquées

Rapport de projet

Traitement de signal

Détection de contour

Table des matières

Introduction.....	3
Définition de la base de données.....	3
Choix des bibliothèques python.....	4
Méthode de détections de contours	4
Méthodes dérivatives.....	4
L'approche Gradient.....	4
Opérateurs dérivatifs du premier ordre.....	6
Opérateur de Sobel.....	6
Code.....	7
Code.....	7
Histogramme des intensités de bord.....	8
Axes d'amélioration du projet.....	9
Conclusion.....	9

Introduction

Dans le cadre de la formation Mathématiques Appliquées dispensé à CY-Tech, nous avons étudié le traitement du signal. Cette matière qui s'inscrit dans la composante mathématiques de notre cursus, nous a permis de comprendre les concepts clés de la manipulation d'images et de signaux. Nous avons notamment travaillé à la création de filtre gaussien et à la création de matrices de gradient.

Pour pratiquer, nous avons été, par ce travail, amené à écrire un programme permettant de trouver les contours des objets présents sur un image, de manière automatique. La détection de contours d'une image a pour principe l'acquisition d'une représentation compacte de toutes les propriétés significatives de l'image perçue. Cette méthode doit être efficace car la validité et l'efficacité ainsi que les possibilités de réalisation des traitement ultérieurs y sont directement liées.

Le sujet de ce projet est l'élaboration d'un algorithme qui permet d'extraire les contours d'une image donnée grâce à la méthode de Sibel.

Définition de la base de données

Pour mener à bien notre projet nous avons procédé à un choix des images à soumettre à notre algorithme pour pouvoir développer rapidement les premières fonctions. L'exécution des algorithmes prend un temps non négligeable, et il était important d'utiliser des petites images pour le début des opération.

Nous avons donc débuter le projet avec 4 petites images de taille raisonnable, puis par la suite de notre projet nous avons tester des images d'une taille supérieure au première. Notre image de référence a été celle là.



Choix des bibliothèques python

Lors de ce projet, pour la réalisation de notre algorithme, nous avons été mené à employer des librairies nécessaire à l'accomplissement de ce travail.

La bibliothèque NumPy qui est une bibliothèque fondamentale pour le calcul numérique en Python. Elle fournit des structures de données telles que les tableaux multidimensionnels qui sont utilisées pour représenter et manipuler les images sous forme de matrices.

La bibliothèque OpenCV (cv2) qui est spécialisée dans la vision par ordinateur. Elle est utilisée ici pour lire une image depuis un fichier, la convertir en niveau de gris, et effectuer des opérations de traitement d'image, telles que la conversion d'une image couleur en niveaux de gris.

La bibliothèque Matplotlib qui est une bibliothèque de traçage en 2D qui permet de visualiser les images, les graphiques, les histogrammes. Dans ce code, elle est utilisée pour afficher l'image résultante, l'histogramme des intensités de bord, ainsi que les matrices de convolution et de corrélation.

Méthode de détections de contours

L'efficacité d'une méthode de détection de contours dépend de l'application visée. En effet, un algorithme de détection de contours conçu, en fonction de l'utilisation ultérieure des contours obtenus et de la précision requise. Plusieurs méthodes de détection de contours existes, dans le cadre de notre projet nous sommes intéressés à la méthodes dérivatives

Méthodes dérivatives

La détection de contours repose sur l'étude de la fonction d'intensité dans l'image. Les points du contour sont obtenus en maximisant les extréma locaux de la norme du gradient ou en détectant les passages par zéro du Laplacien.

Les approches dérivatives sont les plus rapides pour détecter et localiser un signal. On distingue les méthodes basées sur le gradient et celles basées sur le Laplacien. Pour notre projet, nous nous sommes orientés sur l'approche du gradient.

L'approche Gradient

L'approche du gradient permet de détecter les variations locales de la fonction $I(x,y)$ représentant l'intensité.

Un contour d'orientation θ au point (x, y) voir figure 3 est détecté par un maximum de la dérivée directionnelle, dans la direction Φ du gradient $I\nabla(x, y)$, c'est-à-dire par le maximum de la fonction (1-1) :

$$G(\Phi) = |\nabla I(x, y)| \cdot \vec{n} \quad (1-1)$$

\vec{n} Étant le vecteur unitaire dans la direction du gradient : $\vec{n} = (\cos(\Phi), \sin(\Phi))$

La dérivée première directionnelle de $I(x, y)$ suivant n s'écrit alors comme suit :

$$G(\Phi) = |\nabla I(x, y)| \cdot \vec{n} = \cos(\Phi) \frac{\partial I}{\partial x} + \sin(\Phi) \frac{\partial I}{\partial y} \quad (1-2)$$

En dérivant $G(\Phi)$ et en annulant l'expression (1-2), on trouve l'orientation du gradient Φ tel que :

$$\Phi = \arctan \left(\frac{\frac{\partial I}{\partial y}}{\frac{\partial I}{\partial x}} \right)$$

La direction du gradient maximise la dérivée directionnelle, elle est normale au contour et s'oriente de la partie claire vers la partie foncée.

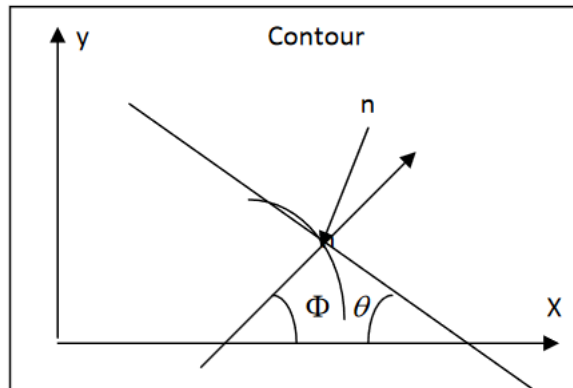


Figure 2: orientation d'un contour

Les points de contour dans une image sont caractérisés par des extrema locaux du gradient. Le principe de la détection de contours par l'utilisation du gradient consiste à calculer d'abord le gradient de l'image dans deux directions orthogonales puis le module du gradient noté G tel que :

$$G = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}$$

Il faut ensuite sélectionner les contours les plus marqués, c'est-à-dire les points qui vérifient $G(x,y) > S$, où S le seuil fixé.

Dans le cas où la norme du gradient au point de contour varie fortement, il n'existe pas de seuil S permettant d'obtenir les vrais points de contours sans sélectionner aussi ceux dus au bruit.

Opérateurs dérivatifs du premier ordre

Dans le cas discret les dérivées directionnelles suivant les directions horizontales et verticales sont approchées par de simples différences finies. Dans une images numériques, I est défini en nombre de pixels. Soir I(x,y) l'image à traiter (x l'indice ligne, y l'indice colonne), les images $I_x[x,y]$ et $I_y[x,y]$ sont de la forme :

$$\frac{\partial I}{\partial x} \approx I_x[x,y] \approx I[x+1,y] - I[x,y]$$

$$\frac{\partial I}{\partial y} \approx I_y[x,y] \approx I[x,y+1] - I[x,y]$$

En terme de filtrage, cela correspond au masque de convolution :

$$- [0 \ -1 \ 1] \text{ pour } \frac{\partial I}{\partial x}$$

$$- [0 \ -1 \ 1]^T \text{ pour } \frac{\partial I}{\partial y}$$

Ainsi, l'image du gradient notée g, peut être obtenue par filtrage linéaire. G est obtenue par convolution de l'image I avec un filtre de masque M :

$$g = I * M$$

L'algorithme de convolution de l'image I avec un filtre de masque M de taille 3*3 pour tout pixel de coordonnées (x,y) :

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

Opérateur de Sobel

On applique les deux masques notés H1 et H2 pour produire les deux gradients directionnels au point (x,y) par convolution

$$\begin{cases} H1 = \frac{1}{4} \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \\ H2 = \frac{1}{4} \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} \end{cases}$$

Code

Nous avons déclaré les matrices de Sobel pour les opérateurs Sobel X et Sobel Y grâce à la bibliothèque numpy.

```
sobel_x = np.array([[ -1,  0,  1], [-2,  0,  2], [-1,  0,  1]]) # Masque pour gx
sobel_y = np.array([[ -1, -2, -1], [ 0,  0,  0], [ 1,  2,  1]]) # Masque pour gy
```

$$\begin{cases} g^1_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-i} \cdot H1_{i,j} \\ g^2_{x,y} = \sum_{i=-1}^1 \sum_{j=-1}^1 I_{x-i,y-i} \cdot H2_{i,j} \end{cases}$$

Code

```
gx = (img[i - 1][j - 1] + 2*img[i][j - 1] + img[i + 1][j - 1]) - (img[i - 1][j + 1] + 2*img[i][j + 1] + img[i + 1][j + 1])
gy = (img[i - 1][j - 1] + 2*img[i - 1][j] + img[i - 1][j + 1]) - (img[i + 1][j - 1] + 2*img[i + 1][j] + img[i + 1][j + 1])
```

Finalement la norme du gradient au point (x,y) est obtenue par $G_{x,y} = \sqrt{g_1^2 + g_2^2}$

Après application de l'algorithme final, on obtient le résultat suivant.



Histogramme des intensités de bord

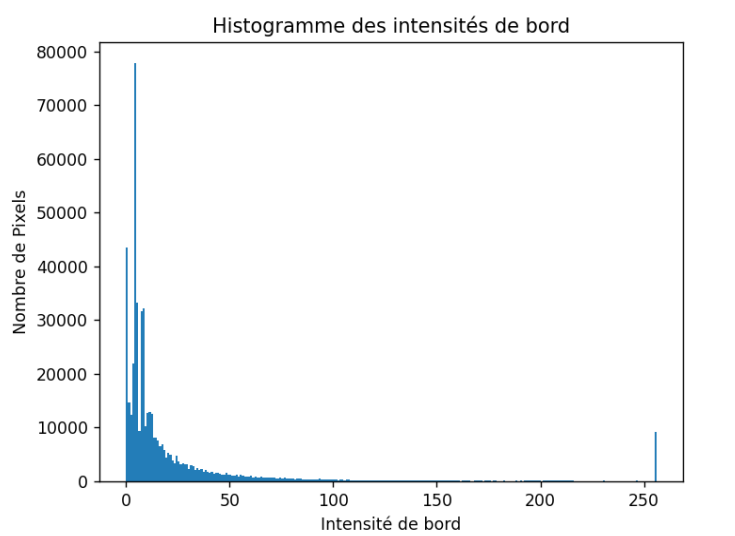
On appelle histogramme de l'image I, la fonction H définie sur l'ensemble des entiers naturels est donnée par l'équation :

$$H(x) = \text{Card} \{P - I(P) = x\}.$$

C'est-à-dire que $H(x)$ traduit le nombre d'apparitions du niveau de gris x dans l'image I. Cette définition se généralise aux images multi-bandes, l'histogramme est alors une fonction de p variables où p désigne le nombre de canaux. L'histogramme est un outil privilégié en analyse d'images car il présente un résumé simple, mais souvent suffisant du contenu de l'image. On distingue 3 types d'histogrammes d'images :

- Histogramme unimodal : Ce type d'histogramme n'a qu'un pic, il représente alors soit un objet soit un fond.
- Histogramme bimodal : Il est formé de deux pics, séparés par une vallée et l'on déduit ainsi qu'il existe un objet sur un fond.
- Histogramme multimodal : Il est formé de plusieurs modes séparés (plusieurs pics séparés par plusieurs vallées) qui nous renseignent sur la présence de plusieurs objets.

Lors de la réalisation de notre projet, nous avons réalisé cet histogramme sur notre image, nous avons obtenu ce graphique :



Nous sommes dans le cas d'un histogramme multimodal ce qui est en adéquation avec notre image d'origine.

Axes d'amélioration du projet

L'algorithme a été testé avec une petite image afin de ne pas avoir un temps de chargement trop important. La complexité principale provient de l'opérateur Sobel, et elle est linéaire par

rapport à la taille de l'image, soit $O(m*n)$ où m est la hauteur de l'image et n la largeur de l'image. Donc si on traite une image de $60*60$ prend environ 10 secondes, traiter une image de $6000*6000$ prendrait simplement 100 fois plus de temps soit 1000 secondes environ 16,67 minutes.

Conclusion

Finalement, notre projet de détection de contours d'objets sur un image en utilisant la méthode de Sobel a été une exploration approfondie du traitement du signal. Nous avons développé un algorithme basé sur les opérateurs dérivatifs du premier ordre, en particulier les opérateurs de Sobel, pour calculer les gradients dans les directions horizontales et verticales.