

Groupe EFREI M1 IRV S8

Module : Traitement d'Image Avancé

Patrick Bonnin

Année 2016 - 2017

Traitement d'Image Avancé : Travaux Pratiques : Détection de Contours

Cet énoncé comporte un rappel de cours¹. Il propose deux implantations : séparée, puis regroupée des procédures :

- *de calcul du gradient selon l'algorithme de Kirsh à 4 directions ;*
- *du seuillage sur la norme du gradient (vu au semestre S7 lors du Module d'Introduction au Traitement d'Image) ;*
- *d'Affinage des Contours .*

1 Introduction

Le but de cette séance de Travaux Pratiques est la compréhension et l'implantation en Langage C, dans l'environnement Logiciel de Traitement d'Images *EdVision*, des premières étapes d'une chaîne d'opérateurs de segmentation en contours comprenant :

- l'algorithme de Kirsch à 4 directions ;
- le seuillage sur la norme du gradient ;
- l'affinage des contours, selon l'algorithme à deux voisins, puis à un seul voisin dans l'implantation où les étapes sont regroupées.

Les trois étapes seront en un premier temps séparées. Elles seront ensuite regroupées.

2 Rappels de Cours

2.1 Opérateur de Kirsh à 4 directions

Le principe de l'algorithme de Kirsh à 4 directions, proposé en 1971 consiste en un premier temps à projeter le vecteur gradient \vec{G} selon 4 directions : O_x , O_y , et les deux bissectrices à $\pm 45^\circ$, en utilisant les 4 vecteurs unitaires : \vec{u}_x , \vec{u}_y , \vec{u}_{45} et \vec{u}_{-45} . Puis, le gradient obtenu, comme approximation du gradient réel, est le vecteur projeté le plus proche, comme l'explique la Figure 1.

Soient \vec{G}_x , \vec{G}_y , \vec{G}_{45} et \vec{G}_{-45} les 4 vecteurs projetés. On obtient le gradient approximé \vec{G} de la manière suivante :

1. De nombreuses parties sont extraites de l'ouvrage : *Les Bases du Traitement d'Image et de la Vision Industrielle et Robotique*, auteur : Patrick Bonnin, aux éditions Lulu.com

$$\begin{aligned}
\|\vec{G}\| &= \text{Max}\{|\vec{G} \cdot \vec{u}_x|, |\vec{G} \cdot \vec{u}_y|, |\vec{G} \cdot \vec{u}_{45}|, |\vec{G} \cdot \vec{u}_{-45}|\} \\
&= \text{Max}\{\|\vec{G}_x\|, \|\vec{G}_y\|, \|\vec{G}_{45}\|, \|\vec{G}_{-45}\|\} \\
\text{Arg}(\vec{G}) &= \begin{cases} 0, & \text{si } \|\vec{G}\| = \|\vec{G}_x\| \\ 6, & \text{si } \|\vec{G}\| = \|\vec{G}_y\| \\ 1, & \text{si } \|\vec{G}\| = \|\vec{G}_{45}\| \\ 7, & \text{si } \|\vec{G}\| = \|\vec{G}_{-45}\| \end{cases}
\end{aligned}$$

En effet, $\vec{G} \cdot \vec{u}_i = \|\vec{G}\| \cdot \|\vec{u}_i\| \cdot |\cos(\widehat{\vec{G}, \vec{u}_i})|$, avec $i = x, y, \pm 45^\circ$.

$\|\vec{u}_i\| = 1$, et $\|\vec{G}\| = G$ est la norme du gradient réel. La projection \vec{G}_i la plus semblable du gradient \vec{G} forme l'angle le plus proche de :

- 0, si les deux vecteurs sont de même sens,
- π , si les deux vecteurs sont en sens contraire.

Dans les deux cas, $|\cos(\widehat{\vec{G}, \vec{u}_i})|$ est le plus proche de 1, soit maximal, ce qui justifie l'algorithme proposé.

Les masques de convolution associés aux vecteurs unitaires \vec{u}_x , \vec{u}_y , \vec{u}_{45} et \vec{u}_{-45} pour l'algorithme de Kirsh à 4 directions sont les suivants :

$$\begin{aligned}
M_h \sim \vec{u}_x &= \frac{1}{3} \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix} & M_v \sim \vec{u}_y &= \frac{1}{3} \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} \\
M_{45} \sim \vec{u}_{45} &= \frac{1}{3} \begin{pmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{pmatrix} & M_{-45} \sim \vec{u}_{-45} &= \frac{1}{3} \begin{pmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}
\end{aligned}$$

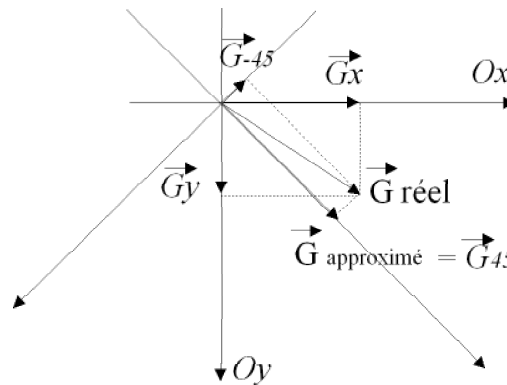


FIGURE 1 – Approximation du gradient \vec{G} : principe de l'algorithme de Kirsh à 4 directions.

Les masques de convolution M_{45} et M_{-45} sont obtenus à partir de M_h et M_v par une rotation de $\pi/4$.

2.2 Seuillage sur la Norme du Gradient

L'opérateur de Seuillage Classique a été programmé au semestre précédent, dans le cadre du module : *Introduction au Traitement d'Images*.

2.3 Affinage des Contours

2.3.1 Position du Problème : Analyse et Principe

Le but de l'opérateur *d'Affinage des Contours* est d'obtenir des (points de) contours d'un pixel d'épaisseur en ne conservant qu'un seul point de contour dans la direction orthogonale au contour.

Sur une image réelle, la transition entre les deux zones homogènes considérées (c'est-à-dire le contour) est rarement "franche". Le profil n'est pas une marche d'escalier parfaite, et génère donc un gradient de norme non nulle sur plusieurs pixels selon la direction orthogonale au contour (ou selon la direction du gradient) comme cela est montré figure 2.

Pour réaliser l'affinage, on ne garde que les pixels sélectionnés par le seuillage sur la norme, dont la norme du gradient est :

Maximale Locale (ie dans le voisinage 3×3 centré) *Directionnelle*, dans la direction du gradient, c'est-à-dire la direction orthogonale au contour.

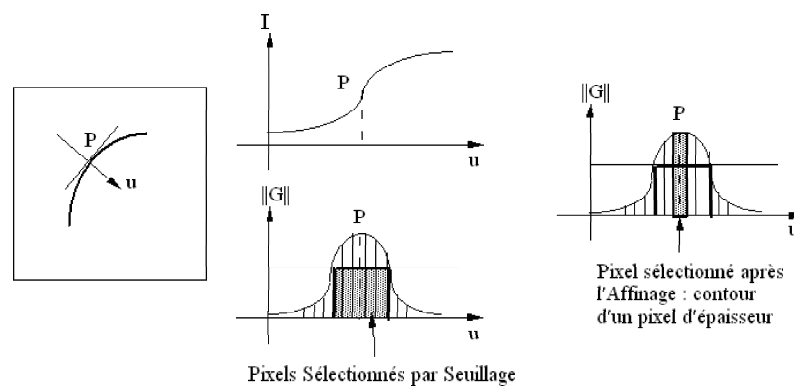


FIGURE 2 – Affinage des contours : analyse du problème.

2.3.2 Principe de l'Affinage à deux voisins

Ainsi, le pixel courant, sélectionné par l'étape de seuillage sur la norme doit être comparé à ses deux voisins V_1 et V_2 dans la direction du gradient. Ces deux pixels voisins sont diamétralement opposés par rapport au pixel courant. L'un est dans le passé, appelons le V_1 , l'autre dans le futur, appelons le V_2 (cf Figure 3).

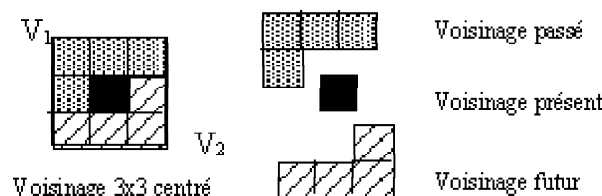


FIGURE 3 – Pixels à comparer dans le voisinage 3×3 .

Le pixel courant est de contour après l'étape d'affinage si et seulement si la norme de son gradient est supérieure à la norme du gradient de ses deux voisins V_1 et V_2 .

L'Algorithme d'Affinage nécessite des disposer en entrée 3 images :

- les images de la norme et de l'argument du gradient,
- l'image des points de contour, obtenue par seuillage sur la norme du gradient.

2.3.3 Algorithme de l'affinage à deux voisins

L'algorithme est proposé table 1

* Balayage Vidéo de l'Image (sauf les bords)

+ Si le Pixel Courant $P(x, y)$ est Point de Contour : ie. $Im_{pts}(x, y) = "1"$

° Recherche des Points Voisins $V_1(x_1, y_1)$ et $V_2(x_2, y_2)$ dans la direction du Gradient $Arg(\overrightarrow{G(x, y)})$ dans le voisinage 3×3 centré de $P(x, y)$

- Si $(\|\overrightarrow{G(x, y)}\| < \|\overrightarrow{G(x_1, y_1)}\|)$ OU $(\|\overrightarrow{G(x, y)}\| < \|\overrightarrow{G(x_2, y_2)}\|)$

@ le Pixel Courant $P(x, y)$ n'est plus Point de Contour : ie. $Im_{pts}(x, y) = "0"$

TABLE 1 – Algorithme d'affinage à deux voisins.

2.4 Le Regroupement des Etapes

2.4.1 Intérêt

L'intérêt de regrouper les calculs des étapes successives d'un même traitement, par exemple :

- calcul du gradient en norme et argument,
- seuillage sur la norme,
- affinage des contours,

est la suppression de parcours des données (ie. de balayages vidéo), qui prennent beaucoup de temps de calcul.

2.4.2 Impératifs

Pour pouvoir regrouper les traitements locaux d'une étape, aux traitements locaux de l'étape précédente, il faut bien évidemment que les données nécessaires aux traitements de cette étape soient disponibles, alors que l'étape précédente n'est pas encore totalement terminée.

Plus précisément, compte tenu de l'implantation en mode "Flot de Données", c'est-à-dire selon le "Balayage Vidéo", les pixels du voisinage 3×3 centré du pixel courant appartenant au **Futur** n'ont pas encore été examinés.

Ainsi, les traitements locaux de l'étape à regrouper aux traitements des étapes précédentes ne doivent pas nécessiter de connaître l'information des traitements des pixels du futur du voisinage du pixel courant, car cette information n'est pas encore disponible.

2.4.3 Détail des Regroupements

L'étape de seuillage sur la norme du gradient ne nécessitant que la connaissance de la valeur de la norme du gradient au pixel courant peut être regroupée sans difficulté à la première étape de calcul des gradients.

2.4.4 Une nécessité : l'Algorithme d'Affinage à un seul voisin

L'étape d'Affinage, si l'on utilise l'algorithme classique comparant la norme du gradient pixel courant aux normes du gradient des deux pixels voisins dans la direction du gradient, ne peut être regroupée, car l'un des deux pixels appartient nécessairement au futur. En revanche, si l'on modifie l'algorithme, en ne comparant la norme du pixel courant qu'à celle du pixel voisin, dans la direction du gradient, mais dans le passé, le regroupement est possible. C'est l'*Algorithme d'Affinage à un voisin*.

3 Travaux à Réaliser

3.1 Version Non Optimisée

3.1.1 Fichiers à Réaliser :

Il s'agira de réaliser les fichiers .c suivants :

1. `DetectionContours.c`
2. `LibKirsh4.c` (et .h)
3. `LibSeuillage.c` (et .h)(à reprendre seulement)
4. `LibAffinage.c` (et .h)

3.1.2 Fichier `DetectionContours.c` :

C'est la partie *Interface de l'Opérateur*. En un premier temps, il faut recenser :

- les images, en terme de fichiers, en entrée, et en sortie,
- les images, en terme de structure de données : entête et données "pixel",
- les différents opérateurs à séquencer, et pour chacun d'eux les images (en terme de structures de données) en entrée, en sortie et même en entrée/sortie.

Ecrire ensuite le fichier, en vous référant au Fichier `EdGradientKirsh4SepMasks.c` fournit. Ce fichier constitue l'interface de l'opérateur calculant la norme du gradient selon l'algorithme de Kirsh à 4 directions. Les paramètres sont :

- en entrée : l'image sur laquelle va être calculé le gradient,
- en sortie : les images de la norme et de l'argument du gradient.

3.1.3 Fichier `LibKirsh4.c` :

C'est l'un des fichiers *Code de l'Opérateur*. Il comporte le calcul du gradient selon l'algorithme de Kirsh à 4 directions.

En s'inspirant du fichier `EdLibGradientKirsh4SepMasks.c` fournit, implantation pédagogique de l'algorithme de Kirsh à 4 directions pour laquelle les calculs des 4 masques de convolution : M_h , M_v , M_{45} et M_{-45} sont séparés :

- écrire l'algorithme détaillé regroupant les calculs des 4 masques,
- réaliser l'implantation en Langage C dans l'Environnement Logiciel *EdEnviTI*.

3.1.4 Fichier `LibSeuillage.c` :

Reprendre le code développé au semestre précédent.

3.1.5 Fichier `LibAffinage.c` :

Planter l'opérateur d'Affinage.

3.2 Version Optimisée : Partie Optionnelle

Il s'agit de créer un nouveau projet comprenant, en plus du fichier des fonctionnalités de l'environnement *EdEnviTI* : `EdUtilities.c` les deux fichiers suivants :

- `DetectionContoursOpt.c`, *interface de l'opérateur* en reprenant `DetectionContours.c`
- `LibKirshSeuilAffinage.c`, regroupant, au sein du même balayage vidéo les étapes de :
 - Calcul du Gradient : norme et argument, avec le regroupement des calculs des 4 masques,
 - le seuillage sur la norme du gradient,

- l’Affinage, MAIS à un seul voisin : celui dans le passé, car le calcul du gradient pour le voisin du voisinage futur n’est pas encore réalisé.

Après avoir proposé l’Algorithme d’Affinage à un seul voisin, réaliser les deux fichiers précédemment mentionnés.

3.3 Comparaison des Résultats

Comparer les résultats obtenus par les deux méthodes. En déduire l’efficacité de l’Affinage à un seul voisin. Proposer une explication.