

Vers les filtres différentiels

23 décembre 2018 – B. COLOMBEL

Nous nous intéressons ici à la détection de contours c'est-à-dire à la détection des lieux de sauts d'intensité. Nous allons commencer par un petit exemple naïf afin de mieux saisir le problème.

Nous allons commencer par une image très simple (figure 1 page 2). Regardons les pixels de plus près.

- À gauche de l'image, tous les pixels sont noirs et qu'à droite de l'image, tous les pixels sont blancs.
- La frontière que nous recherchons est très simple, il s'agit de la frontière entre le noir et le blanc.

1. Quels sont les pixels du contours ?

Quel sera alors l'épaisseur du contour ?

il n'est pas aisé de construire un algorithme qui s'intéresse tantôt au voisin de gauche, tantôt au voisin de droite. On va faire ici un choix qui est le voisin de gauche.

2. On construit une nouvelle image, de même dimension mais donc chaque pixel a pour valeur la différence entre la valeur du pixel (i, j) et celle du pixel $(i - 1, j)$.

(a) Quelles sont les valeurs possibles pour les pixels de la nouvelle image ? Quel problème cela pose-t-il ? Comment le résoudre (simplement) ?

(b) Appliquer notre détecteur de contour naïf aux deux images de la figures 2 page 2.

On ne traitera pas les contours.

(c) Quel problème recontre-t-on ? Proposer une solution.

Notre détecteur naïf est maintenant décomposé en un

- un détecteur de contours horizontaux :

$$I_{ch} = I(i, j) - I(i - 1, j)$$

- et un détecteur de contours verticaux :

$$I_{cv} = I(i, j) - I(i, j - 1)$$

3. On considère l'image de la figure 3 page 2.

(a) Appliquer à cette image le détecteur de contour horizontal.

(b) Appliquer à cette image le détecteur de contour vertical.

(c) L'image finale sera l'image obtenue en calculant la norme euclidienne de pour chaque pixel du vecteur $\begin{pmatrix} I_{ch}(i, j) \\ I_{cv}(i, j) \end{pmatrix}$. Quelle image obtient-on ?

4. Montrer que notre détecteur naïf peut être décomposé en une convolution par un noyau K_x et un noyau K_y avec :

$$K_x = \begin{pmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad \text{et} \quad K_y = \begin{pmatrix} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

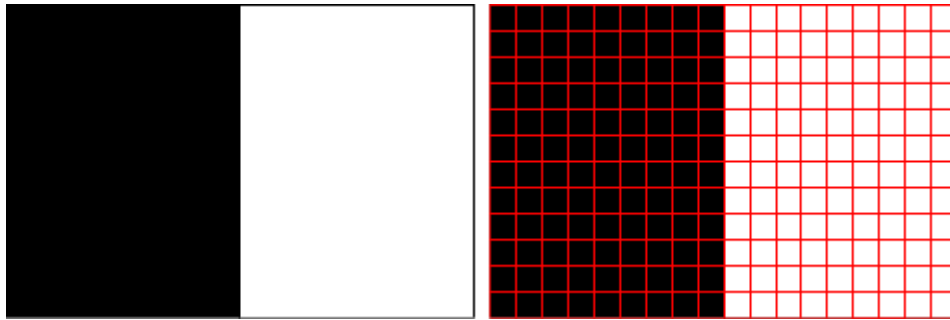


FIGURE 1 – Une image naïve pour comprendre la détection de contour. À droite, un grossissement des pixels.

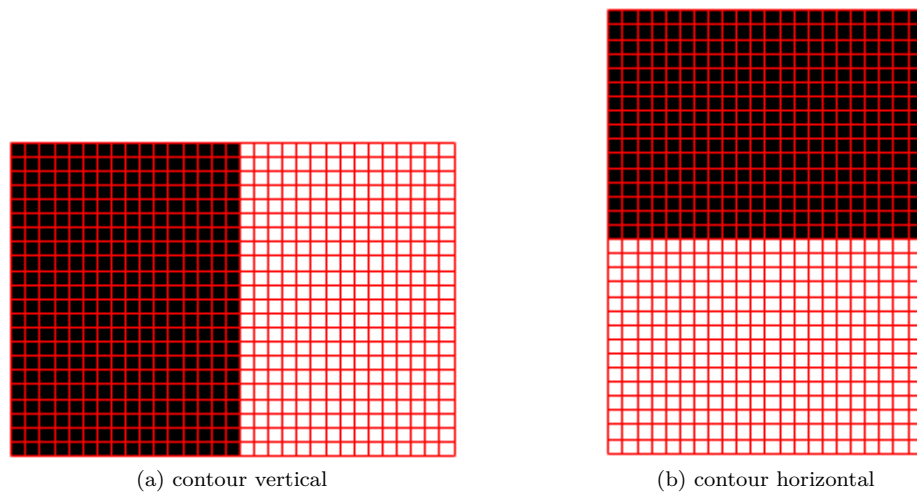


FIGURE 2 – Images naïves

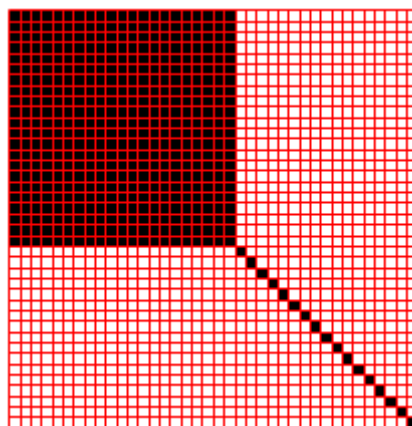


FIGURE 3 – Une image un peu plus complexe