

Série de td n° 3

Exercice 1:

Effectuer la détection de contours de l'image ci-dessous en utilisant les deux masques h_x et h_y et en prenant en compte les informations suivantes :

20	20	30	40	40	40
20	22	30	37	40	40
30	30	30	30	30	30
40	37	30	25	22	22
40	40	30	25	25	25
40	40	30	25	27	20

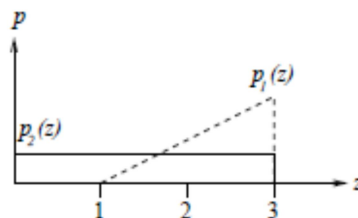
$$h_x = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}.$$

$$h_y = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}.$$

- Les orientations sont approximées à 8 directions : $0, \pm 45, \pm 90, \pm 135, 180$
- Les faux contours sont tous mis à zéro et les contours réels à 1
- Un seuillage final est appliqué à la fin avec un seuil égal à 40

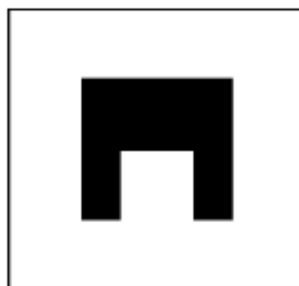
Exercice 2 :

Considérons la figure ci-dessous où $p_1(z)$ correspond à la fonction de densité de probabilité de l'objet et $p_2(z)$ correspond à la densité de probabilité de fond. Calculer le seuil optimal du niveau de gris pour faire une segmentation sachant que le nombre de pixel du fond est deux fois le nombre de pixels de l'objet ($p_2=2p_1$).



Exercice 3 :

Segmenter l'image suivante en utilisant l'algorithme *split and merge*. Utiliser le critère $p(R_i)=vrai$, lorsque tous les pixels dans la région R_i ont la même intensité et construire le Quadtree correspondant.



Exercice 4 :

Considérons deux séquences d'images consécutives :

3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3
3	3	7	3	3	3	3	3	3
3	3	9	7	5	3	3	3	3
3	3	9	9	7	5	3	3	3
3	3	9	9	9	7	5	3	3
3	3	3	3	3	3	3	3	3

Time t_1

3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3
3	3	7	3	3	3	3	3	3
3	3	9	7	5	3	3	3	3
3	3	9	9	7	5	3	3	3
3	3	9	9	9	7	5	3	3
3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3

Time t_2

La fonction d'intensité est $f(x,y,t)$. Nous considérons les pixels de la position (3,4) et (4,4), donc $f(3,4,t_1)=7$ et $f(4,4,t_1)=3$

1. Utiliser les masques de Prewitt 3x3 pour estimer les dérivées spatiales dans les points (3,4, t_1) et (4,4, t_1)
2. Estimer la dérivée temporelle dans les positions (3,4) et (4,4)
3. Supposons que le vecteur flot de l'image $v=(x, y)^T$ est le même pour les pixels voisins. Estimer v en utilisant l'équation du flot de l'image