http://www.mesmaths.com/spip.php?article419

4-contour d'une figure

- SNT - 7-Photographie numérique -

Date de mise en ligne : vendredi 10 mai 2019

ici une activité pour vous aider à finaliser un programme permettant de détecter **les contours d'une figure**

effet obtenus

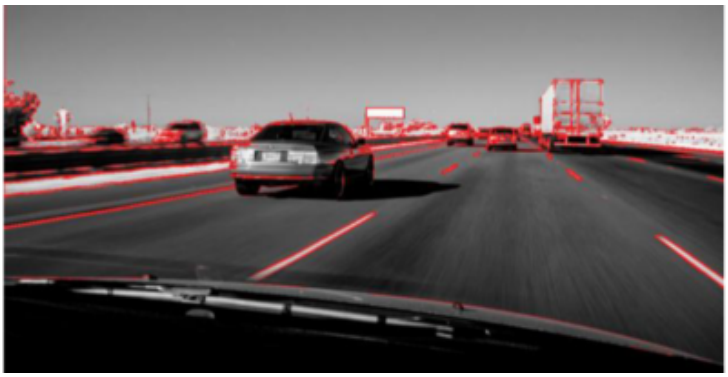
exemple1	exemple2	exemple3

Pourquoi ?

un exemple d'application : la voiture autonome

Quand une voiture autonome est en mouvement, il est important qu'elle sache reconnaître les objets qui l'entourent : véhicules, piétons, cyclistes, les panneaux de signalisation.

Pour suivre la route, elle doit savoir en particulier reconnaître les bordures, les lignes blanches, les trottoirs, etc.



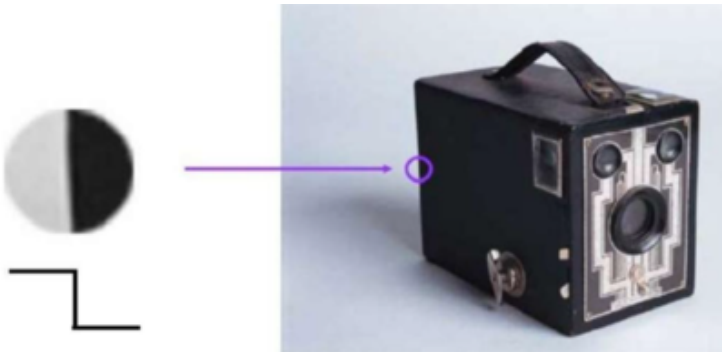
Pour cela, les photos réalisées par ses capteurs sont traitées par des algorithmes qui permettent de détecter les contours des objets qu'elle rencontre.

L'objectif de cette activité est de découvrir le fonctionnement d'un de ces algorithmes de détection de contours.

c'est quoi ?

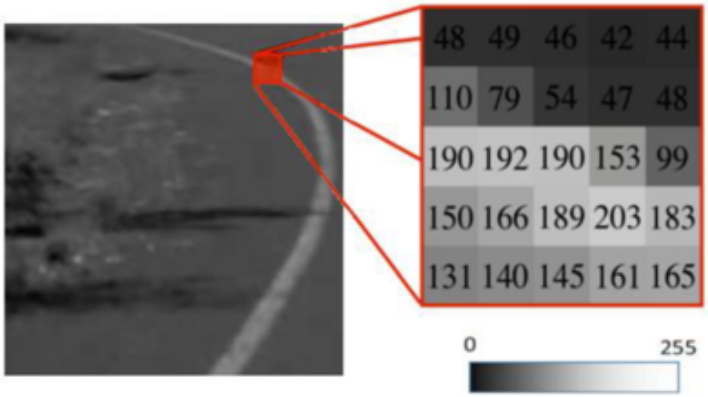
Mais d'abord, comment "définir" un contour ?

Dans certaines situations (comme sur l'image ci-dessous), cela paraît évident de décrire un contour.



Comment formaliser cette 'intuition' ?

question 1 : repérer et sélectionner les pixels appartenant au contour que l'on visualise sur l'agrandissement de l'image en niveaux de gris ci-dessous (noter les niveaux de gris correspondant à ce qu'on estime être le contour de la figure)



réponse

190	192	190		
		189	203	183

question 1 : expliquer comment on peut déterminer si un pixel appartient à un contour.

réponse

Ce sera le cas lorsque ses **pixels voisins** auront des valeurs significativement différentes.

pixels voisins

Pour repérer la position d'un pixel, on repère d'abord la colonne x à laquelle il appartient, puis la ligne y à laquelle il appartient.

Notons un tel pixel P(x, y) et NG(x, y) son niveau de gris.

Pour détecter si le pixel P(x, y) appartient ou non à un contour, une des méthodes consiste à chercher une rupture d'intensité entre 2 pixels symétriques par rapport à P(x, y), appelés **pixel voisins**.

Pour cela on calcule la différence entre les niveaux de gris de 2 pixels voisins. Si cette différence est élevée, le pixel fera parti d'un contour. Sinon, non.

Question 3 : Reproduire le tableau ci-dessous et compléter la position de chaque pixel composant l'image suivante et colorier d'une même couleur les couples de pixels voisins de P(x,y).

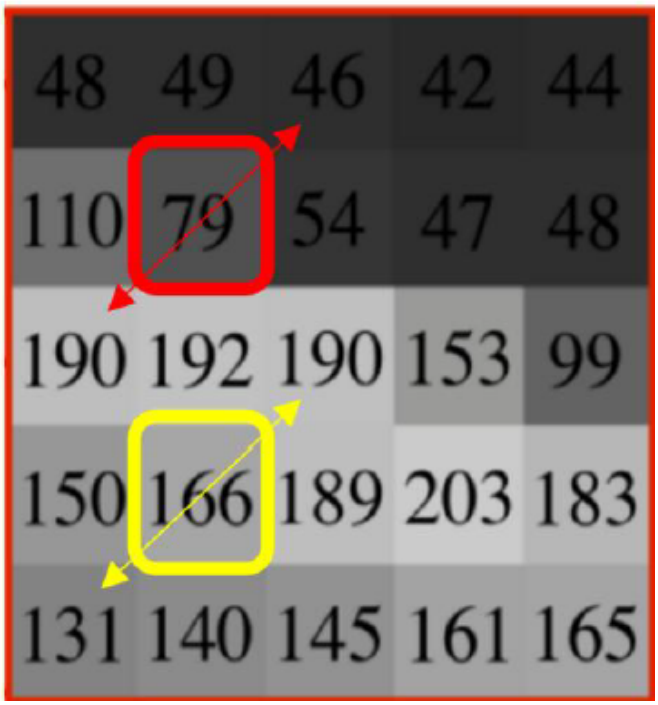
	$P(x, y)$	
		$P(x+1, y+1)$

réponse

P(x-1,y-1)		P(x+1,y-1)
	P(x,y)	
P(x-1,y+1)		P(x+1,y+1)

des calculs

Pour les questions suivantes, on se basera sur la figure suivante :



Question 4 : Quelle est la différence de niveaux de gris entre les 2 pixels voisins du pixel P(2, 2) ? Écrire l'opération effectuée.

réponse

190 - 46 = 144

Question 5 : Même question pour le pixel P(2, 4).

réponse

190 - 131 = 59

Question 6 : Parmi ces deux pixels, lequel semble appartenir à un éventuel contour. Pourquoi ?

réponse

le Pixel P(2,2) car les différences de valeurs entre pixels voisins sont plus grandes.

4 voisins

Avec les exemples précédents, nous n'avons pris en compte qu'un couple de pixels voisins. On ne tient donc compte que d'une unique direction. Pour éviter cela on va donc **prendre en compte 2 couples de pixels voisins** avec des directions opposés (voir schéma).

On calcule ensuite pour chacun des couples de pixels voisins, le carré de la différence des niveaux de gris. Enfin, on ajoute les résultats obtenus.

Question 8 : Calculer cette somme pour P(3, 2) et P(2, 4).

réponse

pour le pixel P(3,2) :

$(192 - 42)^2 + (153 - 49)^2 = 33\ 316$

pour le pixel P(2,4) :

$(190 - 145)^2 + (190 - 131)^2 = 5\ 506$

Question 9 : Exprimer cette somme pour le **pixel P(x, y)**.

réponse

$[P(x-1,y-1) - P(x+1,y+1)]^2 + [P(x+1,y-1) - P(x-1,y+1)]^2 = 33\ 316$

Question 10 : A partir que quel seuil peut-on décider si un pixel fait partie du contour ?

réponse

Ça dépend !!

programme

On souhaite programmer en langage Python un algorithme qui permet d'afficher le contour d'une image.

Vous pouvez tenter d'adapter les programmes précédents ou utiliser celui donné dans le bloc réponse, sachant qu'il y aura **une ligne à compléter** !

Cette ligne à compléter est évidemment en lien avec les questions précédentes, en particulier le fait de décider si un pixel fait partie ou non du contour d'une figure.

réponse

```
"""IMPORTANT : l'image importée devra se trouver dans le même fichier que ce programme. L'image modifiée sera elle aussi enregistrée dans ce dossier"""

 #importation obligatoire pour utiliser la fonction analysant le code RGB pixel par pixel d'une image
 from PIL import Image

 #nom de l'image à saisir A MODIFIER
 im=Image.open("paysage.jpg")
 #prend en compte la largeur de l'image et sa hauteur
 largeur,hauteur=im.size
 #définie une nouvelle image vierge de même dimension
```

```
im_contour=Image.new("RGB",(largeur,hauteur))

# y varie de 1 a  (hauteur-2) et x varie de 1 a  (largeur-2)
for y in range(1,hauteur-1):
for x in range(1,largeur-1):
# p est la valeur RGB du pixel de l'image de départ
p=im.getpixel((x,y))
# a, b, c et d contiennent les valeurs du niveau de gris des pixels voisins  du pixel P(x,y)
a = (im.getpixel((x-1, y-1))[0]+im.getpixel((x-1, y-1))[1]+im.getpixel((x-1, y-1))[2])/3
b = (im.getpixel((x+1, y+1))[0]+im.getpixel((x+1, y+1))[1]+im.getpixel((x+1, y+1))[2])/3
c = (im.getpixel((x-1, y+1))[0]+im.getpixel((x-1, y+1))[1]+im.getpixel((x-1, y+1))[2])/3
d = (im.getpixel((x+1, y-1))[0]+im.getpixel((x+1, y-1))[1]+im.getpixel((x+1, y-1))[2])/3

##on applique la transformation :
# c'est la PARTIE A MODIFIER
formule =
##
# on construit un test : si la valeur obtenue dans la formule est
# supérieure au seuil de 2500, le pixel (x,y) fait parti du contour,
# on le colorie en noir. Sinon, on le colorie en blanc.
if formule > 2500: #seuil modifiable
im_contour.putpixel((x, y), (0,0,0))
else :
im_contour.putpixel((x, y), (255,255,255))

#on nomme la nouvelle image pour l'enregistrer
im_contour.save("paysage_contour.jpg")#nom a modifier
#on visulaise cette nouvelle image
im_contour.show()
```