

VA51 : Compte rendu TP évalué – Valentin MERCY

I) Détermination du point de départ

Question 1: Comme vu dans le TP 1 et étant donné ces informations, calculer les coordonnées de la projection du point P dans l'image acquise par la caméra embarquée sur le véhicule. Détailler vos calculs. Ces coordonnées seront utilisées pour initialiser la deuxième étape de la segmentation par croissance de région.

Avec les coordonnées intrinsèques données en énoncé, on obtient la matrice des paramètres intrinsèques suivante :

$$K = \begin{pmatrix} \alpha_u & 0 & u_0 \\ 0 & \alpha_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 35 \times 10 & 0 & 250 \\ 0 & 35 \times 10 & 135 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 350 & 0 & 250 \\ 0 & 350 & 135 \\ 0 & 0 & 1 \end{pmatrix}$$

On prendra comme repère scène, le même repère que la caméra. La matrice CMS est donc égale à la matrice identité de taille 4x4 et

Le point P a pour coordonnées homogènes, dans ce repère : $P = cX = \begin{pmatrix} 0 \\ 1.3 \\ 5 \\ 1 \end{pmatrix}$

On effectue la projection perspective du point P pour obtenir ses coordonnées dans le plan rétinien :

$$x = pr_p(cX) = \begin{pmatrix} 0/5 \\ 1.3/5 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0.26 \\ 1 \end{pmatrix}$$

On combine ces deux opérations : $u = K * x = \begin{pmatrix} 250 \\ 226 \\ 1 \end{pmatrix}$

On sait donc que le point P a pour projection, dans l'image 2D, le point de coordonnées (250,226).

Question 2: Quelles sont les limites de cette initialisation ? Imaginer (sans l'implémenter) une méthode plus robuste.

Cette initialisation a pour inconvénient de toujours prendre exactement le même point de l'espace 3D. Or, il se peut que ce point n'appartienne pas à la route, dans le cas par exemple d'un virage : dans ce cas, le point de départ considéré pour la segmentation n'appartiendra pas non plus à la route mais à un point situé du côté extérieur du virage (après la route). Ainsi, la segmentation échouera. Autre inconvénient : dans le cas d'un passage piéton (ou d'autres zones distinctives sur la route comme les bandes blanches centrales), si ce point se situe sur une bande blanche, la segmentation permettra seulement de délimiter cette bande blanche et non la route. Une méthode plus robuste serait d'exploiter le nuage de points d'un LIDAR fixé sur la voiture pour choisir dynamiquement le point P dans l'espace 3D situé à l'avant de la voiture, pour ensuite reprendre le principe proposé ici (projection sur l'image de la caméra et segmentation).

II) Croissance des régions

Question 4: Cette approche a l'inconvénient de laisser des « trous » dans la région segmentée, de plus la région segmentée ne tient pas compte des aspérités de la route telles que les bandes blanches qui sont

VA51 : Compte rendu TP évalué – Valentin MERCY

ignorées. De plus, si des formes n'appartenant pas à la route mais ayant un niveau d'intensité proche se trouvent à proximité d'elle sur l'image, celles-ci seront incluses dans le résultat de la segmentation.

III) Post-traitements

Question 5 : Une solution serait ici d'utiliser la morphologie mathématique vue dans le TP3, plus particulièrement la dilatation. En effet, cette solution s'applique parfaitement dans ce cas (image binaire) et permet d'atteindre le but recherché. La dilatation permettra de combler la majorité des trous puisqu'elle permet, contrairement à l'érosion, d'ajouter des pixels à l'image par rapport à leur voisinage grâce à un élément structurant pertinemment choisi.

L'implémentation de cette dilatation est la suivante :

```
import cv2

import numpy as np

im = cv2.imread('seuillee.png', cv2.IMREAD_GRAYSCALE)

im = np.array(im)

structuringElement = np.array([[0,1,0],[1,1,1],[0,1,0]],np.uint8)
filteredImage = cv2.dilate(im,structuringElement)
cv2.imwrite("dilated.png", filteredImage)
```

On obtient l'image suivante (l'image d'entrée étant l'image binarisée comme proposé en cas d'échec à l'implémentation de la croissance des régions) :



VA51 : Compte rendu TP évalué – Valentin MERCY



On peut bien constater que de nombreux trous ont disparu.

IV) Pré-traitements

Question 7 : Le bruit étant mathématiquement une haute fréquence, il suffit d'employer un filtre passe-bas. Afin de ne laisser passer que les basses fréquences. L'implémentation d'un tel filtre fait intervenir la transformée de Fourier pour produire le spectre de l'image d'entrée, sur lequel on applique un masque noir (par exemple ici un carré) au centre de l'image pour masquer les hautes fréquences (concentrées au centre du spectre). On emploie ensuite la transformée de Fourier inverse pour revenir à l'image d'origine nettoyée de son bruit.

L'implémentation est la suivante :

```
from math import log

import cv2
import numpy as np
from math import log10
from scipy import fftpack

def getSpectrum(im_fft):
    """Builds the Fourier spectrum of an image

    Args:
        im_fft (np.array): The source image matrix

    Returns:
        np.array: The spectrum image matrix
    """
    spectrum = np.zeros(im_fft.shape)
    for indexRow, row in enumerate(im_fft):
        for indexPixel, pixel in enumerate(row):
            spectrum[indexRow][indexPixel] = 10*log(abs(pixel),10)
    return spectrum
```

VA51 : Compte rendu TP évalué - Valentin MERCY

```
def centralSquareMask(img, size):  
    """Add a square mask in the center of an image  
  
    Args:  
    img (np.array): The source image matrix  
    size (int): The side length of central square in px  
  
    Returns:  
    np.array: The masked image matrix  
    """  
    leftStart = (np.shape(img)[1]-size)//2  
    upperStart = (np.shape(img)[0]-size)//2  
    img[upperStart:upperStart+size, leftStart:leftStart+size] = 0  
    return img  
  
if __name__ == "__main__":  
    im = cv2.imread('ressources/route1_bruit1.png', cv2.IMREAD_GRAYSCALE)  
    im_fft = np.fft.fft2(im)  
    im_fft = np.fft.fftshift(im_fft)  
    spectrum = getSpectrum(im_fft)  
    filtered = centralSquareMask(im_fft, 10)  
    filteredSpectrum = centralSquareMask(spectrum, 10)  
    #cv2.imwrite('passe_bas.png', filteredSpectrum)  
    inverted = np.fft.ifft2(filtered)  
    inverted = np.abs(inverted)  
    cv2.imwrite('passe_bas.png', inverted)
```

Le résultat obtenu (sur l'image en niveaux de gris) :



Question 8 : La surexposition de l'image a eu pour conséquence de resserrer l'histogramme vers les valeurs hautes. Il faut donc ré-égaliser l'histogramme à l'aide de la méthode vue dans le TP2.

L'implémentation de cette méthode est la suivante :

VA51 : Compte rendu TP évalué - Valentin MERCY

```
import cv2

import numpy as np

im1 = cv2.imread('ressources/route1_bruit2.png', cv2.IMREAD_GRAYSCALE)
im1 = np.array(im1)

imgHeight = np.size(im1, 0)
imgWidth = np.size(im1, 1)
print(imgWidth)
result = np.zeros((imgHeight, imgWidth))
result = np.ascontiguousarray(result, dtype=np.uint8)
nbPixels = np.size(im1,0)*np.size(im1,1)

a = np.min(im1)
b = np.max(im1)

maxPixelValue = 255

for row in range(imgHeight):
    for pixel in range(imgWidth):
        pixelValue = im1[(row,pixel)]
        hc = float(np.count_nonzero(im1<=pixelValue))/nbPixels
        result[(row, pixel)]= round(maxPixelValue*hc)

cv2.imwrite('egalisation.png',result)
```

Le résultat obtenu :

