CS – 532 – A

Janmejay Mohanty

Homework 4

Source Code:

import numpy as np

import open3d as o3d

from IPython.display import Image

import cv2

import math

def Gaussian\_func(image,weight):

sigma = (weight-1)/6

gauss = np.zeros((weight,weight))

for i in range(weight):

for j in range(weight):

gauss[i,j] = math.exp(-((i)\*\*2+(j)\*\*2)/(2\*sigma\*\*2))/(2\*math.pi\*sigma\*\*2)

return cv2.filter2D(image,-1,gauss)

def Suppression\_func(image):

height,breadth = image.shape

result = np.zeros((height,breadth))

for i in range(2,(height - 1)):

for j in range(2,(breadth - 1)):

suppress = image[i-1:i+1, j-1:j+1]

if(image[i,j] == suppress.max()):

result[i,j] = image[i,j]

return result

Dict\_depth = {}

Dict\_rgb = {}

Dict\_gauss = {}

for i in range(3):

Dict\_depth[i] = cv2.imread("problem1/depth" + str(i+1) + ".png")

Dict\_rgb[i] = cv2.imread("problem1/rgb" + str(i+1) + ".png")

Dict\_gauss[i] = cv2.cvtColor(Dict\_rgb[i], cv2.COLOR\_BGR2GRAY)

Kernel\_x = np.array([[-1,0,1],[-1,0,1],[-1,0,1]])

Kernel\_y = np.transpose(Kernel\_x)

print(Kernel\_y)

x\_1 = cv2.filter2D(Dict\_gauss[0],-1,Kernel\_x)

x\_2 = cv2.filter2D(Dict\_gauss[1],-1,Kernel\_x)

x\_3 = cv2.filter2D(Dict\_gauss[2],-1,Kernel\_x)

y\_1 = cv2.filter2D(Dict\_gauss[0],-1,Kernel\_y)

y\_2 = cv2.filter2D(Dict\_gauss[1],-1,Kernel\_y)

y\_3 = cv2.filter2D(Dict\_gauss[2],-1,Kernel\_y)

X\_1 = cv2.filter2D(x\_1,-1,Kernel\_x)

X\_2 = cv2.filter2D(x\_2,-1,Kernel\_x)

X\_3 = cv2.filter2D(x\_3,-1,Kernel\_x)

XY\_1 = cv2.filter2D(y\_1,-1,Kernel\_x)

XY\_2 = cv2.filter2D(y\_2,-1,Kernel\_x)

XY\_3 = cv2.filter2D(y\_3,-1,Kernel\_x)

Y\_1 = cv2.filter2D(y\_1,-1,Kernel\_y)

Y\_2 = cv2.filter2D(y\_2,-1,Kernel\_y)

Y\_3 = cv2.filter2D(y\_3,-1,Kernel\_y)

# Apply Gaussian smoothing to the derivatives using the 5x5 filter

Gauss\_X\_1 = Gaussian\_func(X\_1,5)

Gauss\_X\_2 = Gaussian\_func(X\_2,5)

Gauss\_X\_3 = Gaussian\_func(X\_3,5)

Gauss\_XY\_1 = Gaussian\_func(XY\_1,5)

Gauss\_XY\_2 = Gaussian\_func(XY\_2,5)

Gauss\_XY\_3 = Gaussian\_func(XY\_3,5)

Gauss\_Y\_1 = Gaussian\_func(Y\_1,5)

Gauss\_Y\_2 = Gaussian\_func(Y\_2,5)

Gauss\_Y\_3 = Gaussian\_func(Y\_3,5)

cv2.imshow("X\_1", X\_1)

cv2.imshow("XY\_1", XY\_1)

cv2.imshow("Y\_1", Y\_1)

cv2.imshow("Gauss\_X\_1", Gauss\_X\_1)

cv2.imshow("Gauss\_XY\_1", Gauss\_XY\_1)

cv2.imshow("Gauss\_Y\_1", Gauss\_Y\_1)

cv2.waitKey(0)

a = 0.05

mr = np.multiply(Gauss\_X\_1,Gauss\_Y\_1)

mr = (np.multiply(Gauss\_X\_1,Gauss\_Y\_1) - np.square(Gauss\_XY\_1,Gauss\_XY\_1)) - a\*np.square(Gauss\_X\_1+Gauss\_Y\_1)

cv2.imshow("Rendered Image", mr)

cv2.waitKey(0)

sup = Suppression\_func(mr)

cv2.imshow("Supressed Image", sup)

cv2.waitKey(0)

Output Images:

















