D0957024 楊宏傑 資訊三丁

程式碼

import cv2

import numpy as np

import os

import re

import math

import numpy as np

import matplotlib.pyplot as plt

from mpl\_toolkits.mplot3d import Axes3D

import scipy.special as sp

from numpy import inf

*# 檔案路徑*

paths = ["./test\_datasets/teapot/","./test\_datasets/bunny/"]

for path in paths:

*# 讀取五張圖片*

    files = os.listdir(path)

*# 取出該路徑下的 bmp 檔*

    png\_files = [f for f in files if f.endswith('.bmp')]

    image = []

    for i in range(0,len(png\_files)):

        img = cv2.imread(path + png\_files[i],0)

        img = cv2.normalize(img, None, 0, 255, cv2.NORM\_MINMAX)

        image.append(img)

*# 圖像梯度方向*

    dx = []

    dy = []

    for i in range(0,len(image)):

        x,y = np.gradient(image[i])

        dx.append(x)

        dy.append(y)

*# 定義照明位置和強度*

    file = open(path + "light.txt", "r")

*# 讀取 file 中的每一行*

    list1 = file.readlines()

*# 建立 list2 用於存放三維向量字串，lightlist 用於存放三維向量(數字)*

    list2 = []

    lightlist = []

    for i in range(0,len(list1)):

        list2.append(list1[i])

*# 找出字串中的數字*

        list2[i] = re.findall(r'-?\d+', list2[i])

*# 將檔案編號移出 list 中*

        list2[i].pop(0)

*# 將 list2 裡面的字串轉為數字*

        lightlist.append(list(map(int,list2[i])))

*# 將 list 轉成 array*

    lightlist = np.array(lightlist)

*# print(lightlist)*

    norms = np.linalg.norm(lightlist, axis=1, keepdims=True)

    lightlist = lightlist / norms

*# print(lightlist)*

*# print(lightlist)*

    albedo\_lst = np.zeros(image[0].shape)

    N\_lst = np.zeros(image[0].shape)

    Nx = np.zeros(image[0].shape)

    Ny = np.zeros(image[0].shape)

    Nz = np.zeros(image[0].shape)

    for i in range(image[0].shape[0]):

        for j in range(image[0].shape[1]):

            I = np.zeros([len(image),1])

*# print(I)*

            for x in range(len(image)):

                I[x] = image[x][i][j]

*# print(I)*

            G = np.dot(np.dot(np.linalg.inv(np.dot(lightlist.T,lightlist)),lightlist.T),I).T

*# print(G)*

            norm = np.linalg.norm(G[0])

            if(norm != 0):

                Nx[i][j] = G[0][0] / norm

                Ny[i][j] = G[0][1] / norm

                Nz[i][j] = G[0][2] / norm

            N\_lst[i][j] = G[0][0]\*0.0722+G[0][1]\*0.7152+G[0][2]\*0.2126

*# N\_lst[i][j] = math.sqrt(G[0][0]\*\*2 + G[0][1]\*\*2 + G[0][2]\*\*2) / norm*

*# 算Albedo |N|*

            rho = np.linalg.norm(G[0])

*# print(rho,G[0])*

            albedo\_lst[i][j] = rho

*# for i in range(Nx.shape[0]):*

*#     for j in range(Nx.shape[1]):*

*#         if(Nx[i][j] >= 1):*

*#             print(Nx[i][j])*

*# 控制在0到255間*

    N\_lst = (255-(N\_lst\*0.5 + 0.5)\*255).astype(np.uint8)

    N\_lst = cv2.merge((Nz, Ny, Nx))

    N\_lst = cv2.normalize(N\_lst, None, 0, 255, cv2.NORM\_MINMAX, cv2.CV\_8UC3)

    albedo\_lst = (albedo\_lst/np.max(albedo\_lst)\*255).astype(np.uint8)

*# print(Nx)*

    Nx = 255 - cv2.normalize(Nx, None, alpha=0, beta=255, norm\_type=cv2.NORM\_MINMAX, dtype=cv2.CV\_8U)

    Ny = 255 - cv2.normalize(Ny, None, alpha=0, beta=255, norm\_type=cv2.NORM\_MINMAX, dtype=cv2.CV\_8U)

    Nz = 255 - cv2.normalize(Nz, None, alpha=0, beta=255, norm\_type=cv2.NORM\_MINMAX, dtype=cv2.CV\_8U)

    print(N\_lst)

*#-------------------------------------------------------------------------------*

*# 取 dx 跟 dy 跟 f(x,y)*

    dx = - (Nx / Nz)

    dx = dx - dx[0,0]

    dy = - (Ny / Nz)

    dy = dy - dy[0,0]

    where\_are\_inf = np.isinf(dx)

    dx[where\_are\_inf] = 0

    where\_are\_inf = np.isinf(dy)

    dy[where\_are\_inf] = 0

    row = np.cumsum(dx,axis=0)

    column = np.cumsum(dy,axis=1)

    dz = row + column

    dz[np.isinf(dz)] = 0

    dx = np.round(dx,decimals=4)

    dy = np.round(dy,decimals=4)

    dz = np.round(dz,decimals=4)

    row = np.round(row,decimals=4)

    column = np.round(column,decimals=4)

*# write the file dx , dy*

    file = open(path + 'dx.txt','w')

    file.write("dx = [")

    for i in range(0,dx.shape[0]):

        for j in range(0,dx.shape[1]):

            file.write("{} ".format(dx[i][j]))

        file.write("\n")

    file.write("]\n")

    file.close()

    file = open(path + 'dy.txt','w')

    file.write("dy = [")

    for i in range(0,dy.shape[0]):

        for j in range(0,dy.shape[1]):

            file.write("{} ".format(dy[i][j]))

        file.write("\n")

    file.write("]\n")

    file.close()

    file = open(path + 'dz.txt','w')

    file.write("dz = [")

    for i in range(0,dz.shape[0]):

        for j in range(0,dz.shape[1]):

            file.write("{} ".format(dz[i][j]))

        file.write("\n")

    file.write("]\n")

    file.close()

    file = open(path + 'row.txt','w')

    file.write("row = [")

    for i in range(0,row.shape[0]):

        for j in range(0,row.shape[1]):

            file.write("{} ".format(row[i][j]))

        file.write("\n")

    file.write("]\n")

    file.close()

    file = open(path + 'column.txt','w')

    file.write("column = [")

    for i in range(0,column.shape[0]):

        for j in range(0,column.shape[1]):

            file.write("{} ".format(column[i][j]))

        file.write("\n")

    file.write("]\n")

    file.close()

*# print(dx)*

*# print(dy)*

*# print(dz)*

*#-------------------------------------------------------------------------------*

    cv2.imshow('Albedo', albedo\_lst)

*# 顯示圖片*

    cv2.imshow('Nx', Nx)

    cv2.imshow('Ny', Ny)

    cv2.imshow('Nz', Nz)

*# for i in range(Nx.shape[0]):*

*#     for j in range(Nx.shape[1]):*

*#         if(Nx[i][j] >= 1):*

*#             print(Nx[i][j])*

    cv2.imshow('N', N\_lst)

*# 按下任意鍵則關閉所有視窗*

    cv2.waitKey(0)

    cv2.destroyAllWindows()

*# 寫入不同圖檔格式*

    cv2.imwrite(path + 'Albedo.png', albedo\_lst)

    cv2.imwrite(path + 'Normal.png', N\_lst)

    cv2.imwrite(path + 'Nx.png', Nx)

    cv2.imwrite(path + 'Ny.png', Ny)

    cv2.imwrite(path + 'Nz.png', Nz)

*# cv2.imwrite(path + 'Normal.png', N\_lst)*

*# # 還原影像深度*

*# depth = np.zeros((height, width))*

*# for i in range(height):*

*#     for j in range(width):*

*#         A = np.array([[normals[i,j,0], normals[i,j,1]], [normals[i,j,1], -normals[i,j,0]]])*

*#         b = np.array([-normals[i,j,2], albedo[i,j]/np.pi])*

*#         x = np.linalg.solve(A, b)*

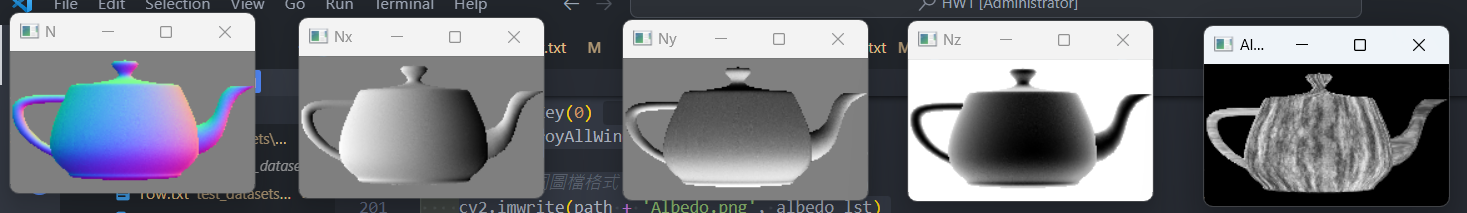
*#         depth[i,j] = x[0]/x[1]*

*# # 可視化結果*

*# cv2.imshow*

實作圖片

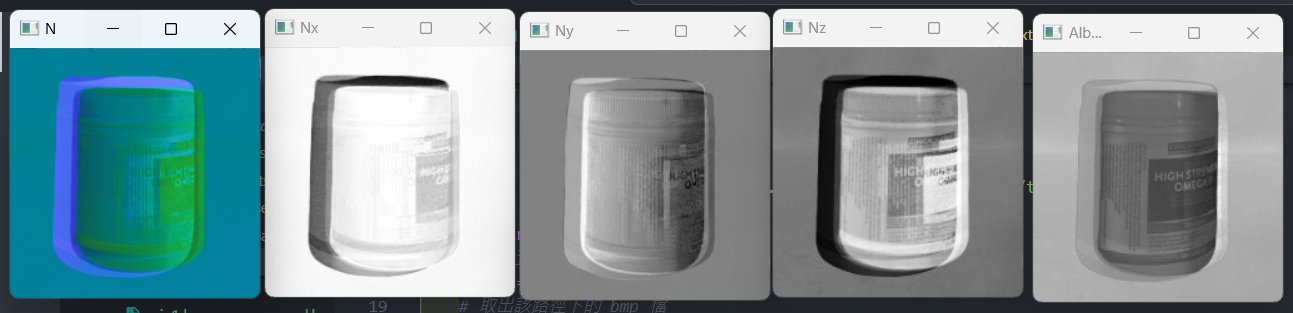
順序(Normal, Nx, Ny, Nz, Albedo)



一張含有 文字, 貓, 屏幕、螢幕, 螢幕擷取畫面 的圖片

自動產生的描述

(Drug)



過程中雖然一開始不是很懂但是上網查了一些資料後，進度就很快，前面的Normal一下子就算出來了，但是要算gx跟gy時就出現了蠻大的問題，而透過老師的講解後，就大致上了解如何處理。而我依照公式的方法，經過了2個禮拜的四天周末，加上每天晚上的努力，我還是未能完整的把物件重構，聽老師說可能是因為gx跟gy取錯方向，但是試了很多遍野都是一樣的結果，所以目前還是放棄了。  
一張含有 圖表 的圖片

自動產生的描述一張含有 圖表 的圖片

自動產生的描述

圖、dx 圖、dy

一張含有 圖表 的圖片

自動產生的描述一張含有 資料表 的圖片

自動產生的描述

圖、dx 圖、dy

Github連結 : <https://github.com/qwe8496516/Computer_Version>

在Branch – master 如果老師有空的話想請老師幫我看一下到底是錯在哪裡