from types import new\_class

import numpy as np

import sys

import cv2

from PIL import Image

def rank\_transform(image, win\_size):

trim = int((win\_size - 1) / 2)

w, h = image.shape

rank = np.zeros((w, h), dtype="int64")

# print(w,h)

for i in range(0, w):

for j in range(0, h):

for x in range(i - trim, i + trim):

for y in range(j - trim, j + trim):

if (0 <= x < w and 0 <= y < h):

if (image[x][y] < image[i][j]):

rank[i][j] += 1

return rank

def disp\_dict(img, win\_size):

trim = int((win\_size - 1) / 2)

w, h = img.shape

# print(w,h)

dict = {}

for i in range(0, w):

for j in range(0, h):

tup = (i, j)

arr = np.zeros((win\_size, win\_size), dtype="int64")

for x in range(-trim, trim):

for y in range(-trim, trim):

if (0 <= x + i < w and 0 <= y + j < h):

arr[x + trim][y + trim] = img[i + x][j + y]

dict[tup] = arr

return dict

def getSad(arr1, arr2):

l = len(arr1)

sub = np.subtract(arr1, arr2)

# print("1:",arr1,"2:",arr2,sub)

total = 0

for i in range(l):

for j in range(l):

total += abs(sub[i][j])

return total

def disp(w, h, dict1, dict2, dir):

# dir = left for left disparity map

# dir = right for right dipsrity map

# w, h = img.shape

# print(w,h)

disp\_map = np.zeros((w, h), dtype="uint8")

for i in range(0, w):

for j in range(0, h):

arr1 = dict1[(i, j)]

best = 0

for d in range(64):

jd = j - d

if (dir == 'right'):

jd = j + d

if (jd >= 0 and jd < h):

arr2 = dict2[(i, jd)]

sad = getSad(arr1, arr2)

# print("11:",arr1,"22:",arr2)

# print(arr1,arr2,sad)

if (d == 0 or sad < best):

disp\_map[i][j] = abs(d)

best = sad

if (sad == 0):

disp\_map[i][j] = abs(d)

break

return disp\_map

def error\_rate(image, disp):

image\_array = np.asarray(image)

dis\_array = np.asarray(disp)

w, h = dis\_array.shape

total\_pix = w \* h

bad\_pix = 0

for i in range(w):

for j in range(h):

div\_f = round(image\_array[i][j] / 4)

if dis\_array[i][j] - div\_f > 1:

bad\_pix += 1

elif dis\_array[i][j] - div\_f < -1:

bad\_pix += 1

error = float(bad\_pix / total\_pix)

error \*= 100

print("Error Rate: " + str(error) + "%")

def pkrn\_disp(w, h, disp, dict1, dict2, dir):

# dir = left for left disparity map

# dir = right for right dipsrity map

disp\_map = np.zeros((w, h))

for i in range(0, w):

for j in range(0, h):

arr1 = dict1[(i, j)]

best = getSad(arr1, dict2[(i, j)])

best2 = best

for d in range(64):

jd = j - d

if (dir == 'right'):

jd = j + d

if (jd >= 0 and jd < h):

arr2 = dict2[(i, jd)]

sad = getSad(arr1, arr2)

if (sad < best):

# disp\_map[i][j] = abs(d)

best2 = best

best = sad

if (sad < best2 and sad > best):

best2 = sad

if (best != 0):

# print(best2/best)

disp\_map[i][j] = (best2 / best) \* 4

else:

disp\_map[i][j] = 256

# print(disp\_map)

med = np.median(disp\_map)

# sparse\_map = np.zeros((w,h))

for i in range(0, w):

for j in range(0, h):

if (disp\_map[i][j] < med):

disp[i][j] = 0

return disp

def main():

image1 = Image.open("disp2.pgm")

left = Image.open("teddyL.pgm")

right = Image.open("teddyR.pgm")

h, w = image1.size

# image1.show()

# left.show()

# right.show()

# apply rank transform to left and right images

print("Starting Rank Transform...")

l\_rank = rank\_transform(np.array(left), 5)

r\_rank = rank\_transform(np.array(right), 5)

print("Finished Rank Transform")

print("Starting 3\*3 Disparity Map...")

ldict = disp\_dict(l\_rank, 3)

rdict = disp\_dict(r\_rank, 3)

ldisp = disp(w, h, ldict, rdict, 'left')

rdisp = disp(w, h, rdict, ldict, 'right')

dm1 = Image.fromarray(ldisp)

dm1.show()

dm2 = Image.fromarray(rdisp)

dm2.show()

error\_rate(image1, dm1)

error\_rate(image1, dm2)

print("Finished 3\*3 Disparity Map")

print("Starting 15\*15 Disparity Map...")

ldict2 = disp\_dict(l\_rank, 15)

rdict2 = disp\_dict(r\_rank, 15)

ldisp2 = disp(w, h, ldict2, rdict2, 'left')

rdisp2 = disp(w, h, rdict2, ldict2, 'right')

dm3 = Image.fromarray(ldisp2)

dm3.show()

dm4 = Image.fromarray(rdisp2)

dm4.show()

error\_rate(image1, dm3)

error\_rate(image1, dm4)

print("Finished 15\*15 Disparity Map")

print("Starting PKRN Disparity Map...")

pkrn\_ldisp = pkrn\_disp(w, h, ldisp2, ldict, rdict, 'left')

pkrn\_dm1 = Image.fromarray(pkrn\_ldisp)

pkrn\_dm1.show()

error\_rate(image1, pkrn\_dm1)

pkrn\_rdisp = pkrn\_disp(w, h, rdisp2, rdict, ldict, 'right')

pkrn\_dm2 = Image.fromarray(pkrn\_rdisp)

pkrn\_dm2.show()

error\_rate(image1, pkrn\_dm2)

print("Finished PKRN Disparity Map")

if \_\_name\_\_ == "\_\_main\_\_":

main()



3x3(left)



3x3(right)



15x15(left)



15x15(right)



PKRN(left)



PKRN(right)

Starting Rank Transform...

Finished Rank Transform

Starting 3\*3 Disparity Map...

Error Rate: 69.95792592592592%

Error Rate: 78.60385185185184%

Finished 3\*3 Disparity Map

Starting 15\*15 Disparity Map...

Error Rate: 21.67822222222222%

Error Rate: 44.3437037037037%

Finished 15\*15 Disparity Map

Starting PKRN Disparity Map...

Error Rate: 51.42933333333334%

Error Rate: 65.8394074074074%

Finished PKRN Disparity Map