透视变换实验报告

07111701班 1120172150 谢威宇 2020.1.2

实验环境

- Windows 10
- PyCharm 2019.2.2
- Python 3.7.4
- numpy 1.17.2
- OpenCV 4.1.2

实验描述

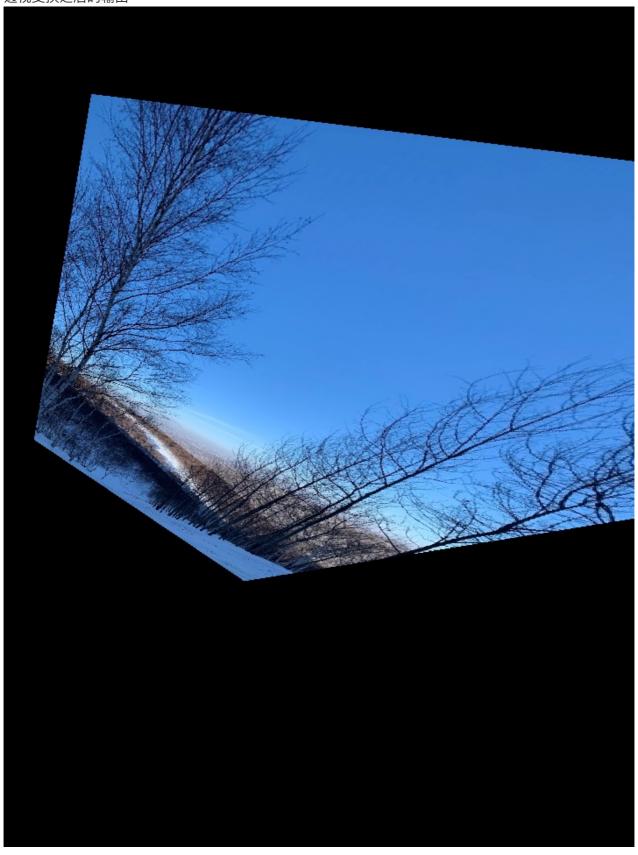
通过矩阵运算进行透视变化。给定一个透视变换矩阵,本程序就可以进行透视变换。本实验实现了对多通道彩色图片的处理。

实验代码

```
import cv2 as cv
import numpy as np
def get_transform_matrix():
   # 返回一个透视变换矩阵
   return np.array([
       [1.4, 0, 100],
       [0, 0.7, 100],
       [0.002, -0.001, 1]
    1)
def perspective_transform(path='test.jpg'):
   # 读取图片
   im = cv.imread(path).astype(np.float64) / 255
   # 获取透视变换矩阵
   tm = get_transform_matrix()
   # 变换矩阵求逆
   itm = np.linalg.inv(tm)
   # 获取新图片的每一个点的坐标
   cc, rc = np.meshgrid(np.arange(im.shape[1]), np.arange(im.shape[0]))
   co = np.vstack((rc.flatten(), cc.flatten(),
np.ones(rc.flatten().shape[0])))
```

```
# 进行逆变换
   co = itm @ co
   co /= co[2]
   # 删除 w
   co = np.delete(co, 2, axis=0)
   # 最近插值法
   co = np.around(co).astype(np.int)
   # 判断采样点是否在图片内部
   choice = np.all([co[0] >= 0,
                    co[0] < im.shape[0] - 1,
                    co[1] >= 0,
                    co[1] < im.shape[1] - 1], axis=0)
   # 限制采样点坐标
   co[np.where(co < 0)] = 0
   co[0][np.where(co[0] > im.shape[0] - 1)] = im.shape[0] - 1
   co[1][np.where(co[1] > im.shape[1] - 1)] = im.shape[1] - 1
   # 图片外部的采样点设为黑色
   nim = np.where(choice, im[co[0], co[1]].T, 0).T
   # 变换成图片尺寸
   nim = nim.reshape(im.shape)
   # 展示图片
   cv.imshow('perspective_transform', im)
   cv.waitKey(0)
   cv.imshow('perspective_transform', nim)
   cv.waitKey(0)
   cv.imwrite('out.jpg', im)
perspective_transform()
```

实验结果



很好地完成了透视变换的实验要求。