数字图像处理第二次作业

吕天
2018011391
计算机科学与技术系

日期：2020 年 3 月 14 日

1 总体概述

本次作业实现了图像扭曲 (Image warping) 的内容，具体而言完成了投影变换、球面变换等课程上讲述的内容，并且自行设计变换函数实现了漩涡效果的变换。在本次作业的完成过程中，未曾与其他人有过关于作业内容的交流，仅仅参考了课件中的相关内容，并且仅仅使用了 python 中的 math、matplolib、numpy 库用于相关的计算以及文件的存取，保证**所有代码均由本人独立完成**，特此声明。所有代码可以在https://github.com/halo2718/DIP上获取。

2 投影变换

根据课件中示例的要求，需要选取四个点作为源图嵌入目标图的位置，不妨设四个点为 $(r_1, c_1), (r_2, c_2), (r_3, c_3), (r_4, c_4)$，则变换可以形式化地表示为

$$(0,0) \rightarrow (r_1, c_1)$$
$$(R,0) \rightarrow (r_2, c_2)$$
$$(0,C) \rightarrow (r_3.c_3)$$
$$(R,C) \rightarrow (r_4, c_4)$$

其中，R, C 分别表示源图的列数和行数。应用齐次坐标的方法，则可以表示为以下的形式:

$$\begin{bmatrix} r_1 & r_2 & r_3 & r_4 \\ c_1 & c_2 & c_3 & c_4 \\ 1 & 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \begin{bmatrix} 0 & R & 0 & R \\ 0 & 0 & C & C \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

利用 python 中的 numpy 库可以求解出变换矩阵的具体结果，其代码列举如下，其中各个点的具体数值坐标是通过 windows 中的画图软件获取的，TR 即为所求的变换矩阵:

```
A = np.array([[0,524,0,524],[0,0,699,699],[1,1,1,1]])
B = np.array([[195,315,264,389],[192,169,536,508],[1,1,1,1]])
r1 = np.linalg.pinv(A)
TR = np.dot(B,r1)
```

由此可得变换矩阵为

$$\begin{bmatrix} 0.2338 & 0.1023 & 19.38 \\ -0.04866 & 0.4856 & 19.33 \\ 1.084 \times 10^{-18} & 7.589 \times 10^{-19} & 1 \end{bmatrix}$$

因此对于目标图中的每个像素点，应用上述投影矩阵的逆矩阵乘以该像素对应的齐次坐标，如果变换得到的点在源图的有效范围内 ($[0, R-1] \times [0, C-1]$)，则将原像素赋值为源图中变换后的像素，由此可以获得题目所要求的图片，如下图 1 所示：

图 1: 投影变换所得的嵌入图片

其中关键代码如下所示：

```
res = np.dot(trans, np.array([i,j,1]).T)
x = int(res[0])
y = int(res[1])
```

3 球面变换

根据课程中介绍的球面变换，则有对于一个 $R_o \times C_o$ 的输出图片，其中的每个像素点的坐标用 (r_o, c_o) 表示，那么就有以下的变换式子：

$$r_i = \frac{1}{\pi} \max(R_i, C_i) \sin^{-1}\left(\frac{\sqrt{r_o^2 + c_o^2}}{\frac{1}{2}\min(R_o, C_o)}\right) \sin\left(\tan^{-1}\left(\frac{r_o}{c_o}\right)\right)$$

$$c_i = \frac{1}{\pi} \max(R_i, C_i) \sin^{-1}\left(\frac{\sqrt{r_o^2 + c_o^2}}{\frac{1}{2}\min(R_o, C_o)}\right) \cos\left(\tan^{-1}\left(\frac{r_o}{c_o}\right)\right)$$

上式中 (r_i, c_i) 代表变换后图像中 (r_o, c_o) 的原像，并且原图的大小为 $R_i \times C_i$。上式中将输出图片的像素放在等式右边，而输入图片的像素放在等式左边是为了符合图像的生成过程：即对于输出图像中每个位置的像素点，通过上述公式可以确定原像的位置，并且在输出图像的位置赋值为对应输入像素的属性值。经过上述操作之后得到的图像如图 2 所示，其中输出的长和宽分别设置为 300 像素。

图 2: 球面变换后得到的图片

可以看出在这种条件下计算出来的球面变化忽略了一些长边的信息，因此为了使得变换后的图片的仍然能够包含长边的信息，因此需要对长边对应的维度乘以修正项，以列数较大时为例，则上述公式变成了：

$$c_i = \frac{1}{\pi}\max(R_i, C_i)\sin^{-1}\left(\frac{\sqrt{r_o^2 + c_o^2}}{\frac{1}{2}\min(R_o, C_o)}\right)\cos\left(\tan^{-1}\left(\frac{r_o}{c_o}\right)\right)\frac{\max(R_i, C_i)}{\min(R_i, C_i)}$$

应用上述修正后得到的球面变换图片如图 3 所示

图 3: 球面变换后得到的图片（包含修正项）

实现该过程的核心代码如下所示：

```
angle = math.atan2(x,y)  ##计算角度
if (x**2+y**2)**(0.5)< r_0: ##判断距离是否有效
    dist = 2*d_0*math.asin((x**2+y**2)**(0.5)/outHeight//2)/math.pi
```

```
if mode == "square":     ##不进行伸缩修正
    X = int(dist*math.sin(angle))
    Y = int(dist*math.cos(angle))
elif mode == "modify":   ##进行伸缩修正
    if row<col:
        X = int(dist*math.sin(angle)*col/row)
        Y = int(dist*math.cos(angle))
    elif row>=col:
        X = int(dist*math.sin(angle))
        Y = int(dist*math.cos(angle)*row/col)
output[i][j]=img[X+cx][Y+cy]     ##赋值，注意加上中心点坐标
```

4 “漩涡”变换

欲实现的效果为以图片几何中心为漩涡中心点，其余像素向某一方向围绕中心点旋转一定角度，并且旋转角度随着到中心点距离增加而成比例地增加，因此可以设计出如下的映射函数，字母的使用与第二部分中保持一致：

$$r_i = \sqrt{r_o^2 + c_o^2}\sin(tan^{-1}(\frac{c_o}{r_o}) + k\sqrt{r_o^2 + c_o^2})$$

$$c_i = \sqrt{r_o^2 + c_o^2}\cos(tan^{-1}(\frac{c_o}{r_o}) + k\sqrt{r_o^2 + c_o^2})$$

式子中的 k 为旋转参数，其值越大则图像旋转得越为剧烈。将输入坐标写在等式左边说明是通过输出图片中的一点来寻找原像并赋值，同时需要保证原像处于有效的范围，将输出图片的像素大小与原图保持一致，则有如下的原图、输出结果分别如图 4，5 所示：

图 4: 原图

图 5: 漩涡变换之后的图片

该操作的核心代码如下所示：

```
radian = math.atan2(y,x)     ##计算角度
radius = int(math.sqrt(x**2+y**2))  ##计算距离
X = int(radius*math.cos(radian+0.002*radius))+cx     ##变换式
Y = int(radius*math.sin(radian+0.002*radius))+cy
X = min(width-1, max(0,X))       ##保证输出有效
Y = min(height-1, max(0,Y))
output[j][i] = img[X][Y]/255     ##赋值
```

5 遇到的问题

1. 在三个变换的编程实现过程中，有时候会把行列、xy、ij 关系弄混导致输出图片错误。
2. 在自行设计变换效果时，推导合理的变换式花费了一定时间。