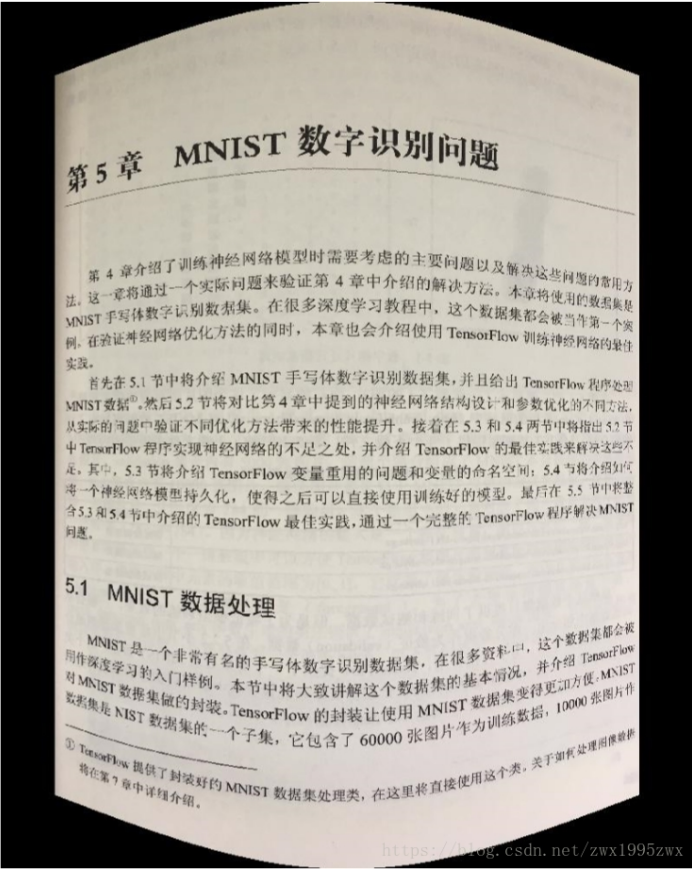
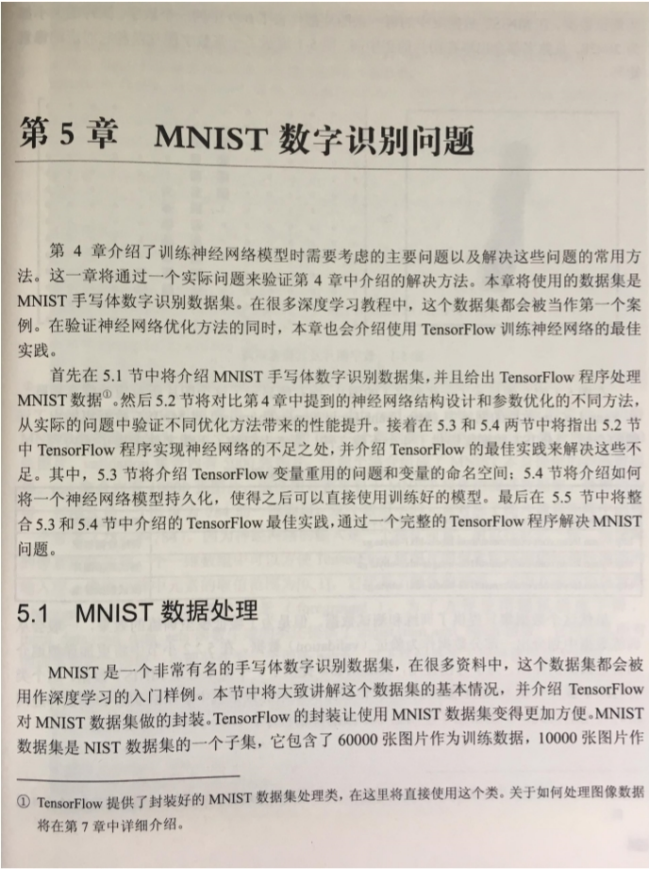


柱面投影介绍与python实现（一）

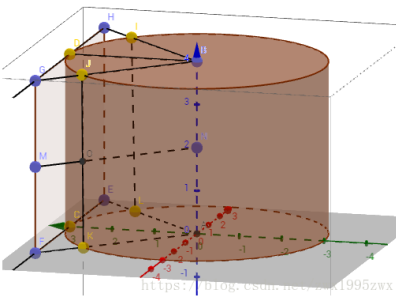
简介

本文主要介绍柱面投影变换，将这种变换用于图像处理，可以产生图像扭曲的效果，如下图所示，产生了将平面图像投影到了柱面上的视觉效果。博客最后给出了柱面投影的python实现，供读者参考。



数学原理

如下图所示，矩形GHEF为待处理的原图，投影到柱面上之后则变成了曲面JDILCK。我们假设相机模型为针孔模型，且所处位置为N点，下面分析原图GHEF与投影曲面JDILCK各像素之间的对应关系。

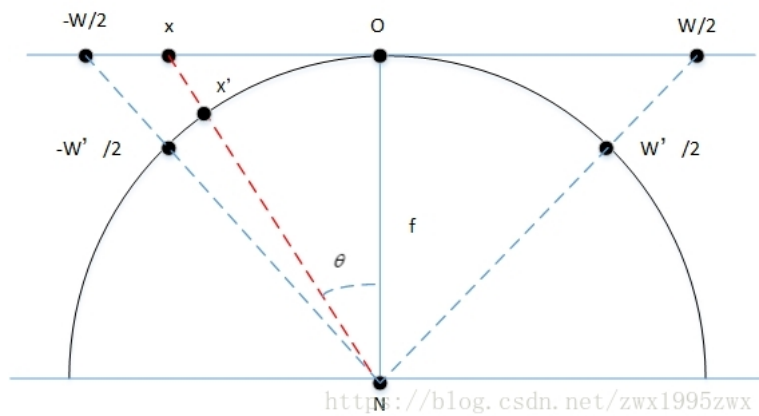


俯视图如下图所示，我们以原图像的中心（W/2, H/2）为原点，假设相机的焦距为f（f的值我们可以随意选择，不需要相机的真实焦距）。

下面我们进行几何分析，点 $-W/2$ 映射到 $-W'/2$ ，点 $W/2$ 映射到 $W'/2$ ，因此直线 $[-W/2, W/2]$ 为原图X方向，曲线 $[-W'/2, 0, W'/2]$ 为投影曲面的X方向。

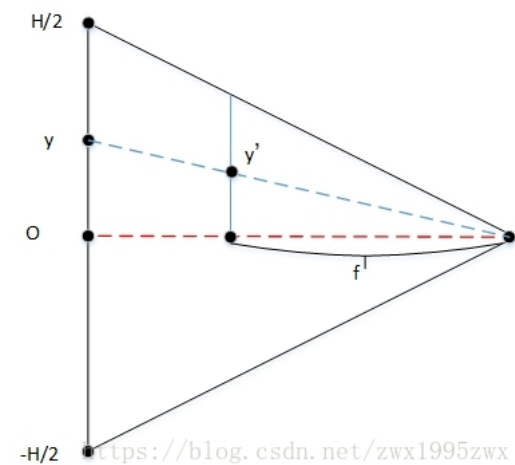
假设原平面上有一点 x ，映射到曲面上为 x' （在投影曲面X方向上的坐标为 x' ，即弧长为 x' ），设角 xNO 为 θ ， $\theta = \arctan$

(x/f) ，弧长 $x' = f \times \theta$ 。因此X方向上柱面投影的映射关系为 $x' = f \times \arctan(x/f)$ 。



接下来看侧视图，以确定Y方向上原图与投影图像素的对应关系。侧视图中的红色虚线与俯视图中的红色虚线相对

应。由俯视图可得线段 $xN = f / \cos(\theta)$ ，由侧视图中的相似三角形关系可得 $y' / f = y / (f / \cos(\theta))$ 。



综上所述，可得原图与投影曲面各像素的对应关系如下

1、 $\theta = \arctan\left(\frac{x}{f}\right)$

2、 $x' = f \times \theta$

3、 $\frac{y}{f \cos(\theta)} = \frac{y'}{f}$

根据上面三个式子，我们便可以对图片进行柱面投影变换了。python实现代码如下所示。

```
1 from skimage.io import imread, imshow,imsave
2 from skimage.transform import resize
3 import math
4 import numpy as np
5
6 img = imread('img.jpg')
7 img = (resize(img , [1000,800])*255).astype(np.uint8)
8
9 ###圆柱投影
10 def cylindrical_projection(img , f) :
11     rows = img.shape[0]
12     cols = img.shape[1]
13
14     #f = cols / (2 * math.tan(np.pi / 8))
15
16     blank = np.zeros_like(img)
```

```

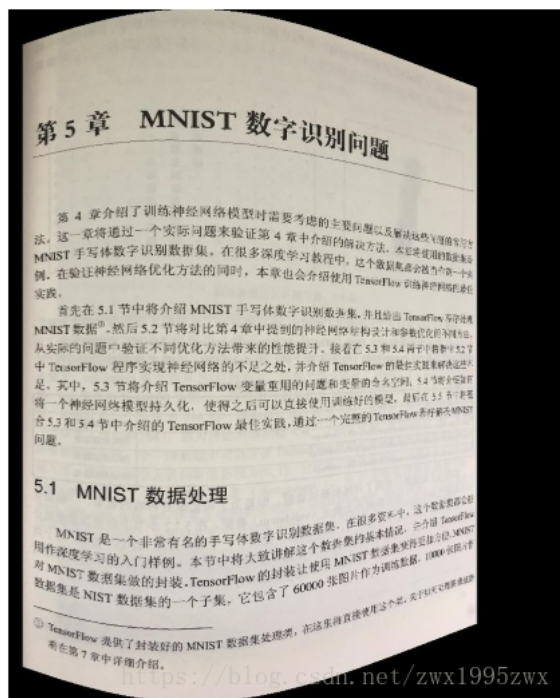
17 | center_x = int(cols / 2) | 18 | center_y = int(rows / 2)
19 |
20 | for y in range(rows):
21 |     for x in range(cols):
22 |         theta = math.atan((x- center_x )/ f)
23 |         point_x = int(f * math.tan( (x-center_x) / f) + center_x)
24 |         point_y = int( (y-center_y) / math.cos(theta) + center_y)
25 |
26 |         if point_x >= cols or point_x < 0 or point_y >= rows or point_y < 0:
27 |             pass
28 |         else:
29 |             blank[y , x, :] = img[point_y , point_x ,:]
30 |     return blank
31 |
32 | waved_img = cylindrical_projection(img,500)
33 | imshow(waved_img)

```

补充:

1、从上面的程序我们可以看出，程序中对三个公式的应用是反的。这是因为程序中两个for循环中的x与y对应的是柱面投影的曲面，point_x与point_y对应的是原图中的坐标点。

2、此外，我们可以将相机的位置N点进行移动，产生不同的柱面变换。在程序实现时，我们直接更改center_x与center_y的值即可。例如，center_x = int(cols/3), center_y = int(rows/2), 产生的效果图如下所示。



zwx1995zwx

关注



6



3



19



打赏

相关推荐

[关于我们](#) [招贤纳士](#) [广告服务](#) [开发助手](#)  400-660-0108  kefu@csdn.net  [在线客服](#) 工作时间 8:30-22:00

公安备案号11010502030143 京ICP备19004658号 京网文〔2020〕1039-165号 [经营性网站备案信息](#) [北京互联网违法和不良信息举报中心](#)
[网络110报警服务](#) [中国互联网举报中心](#) [家长监护](#) [Chrome商店下载](#) ©1999-2021北京创新乐知网络技术有限公司 [版权与免责声明](#) [版权申诉](#)
[出版物许可证](#) [营业执照](#)