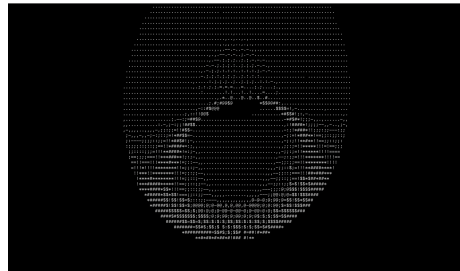
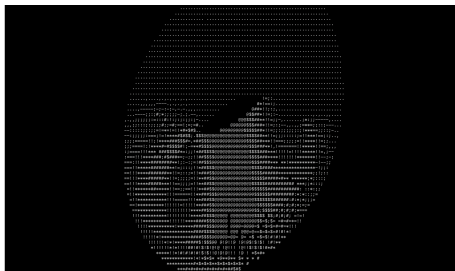
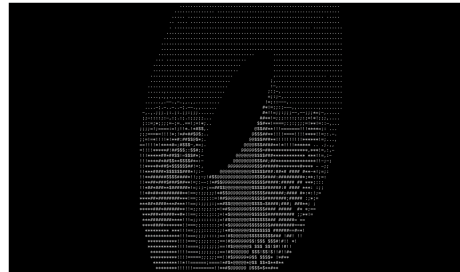
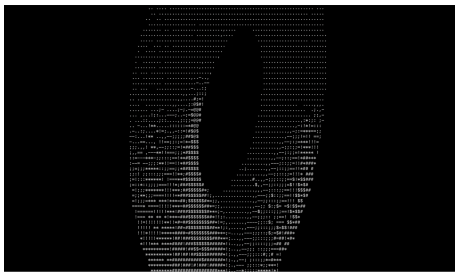


3d 影像显示报告

徐世桐

1 程序目标

在终端内显示旋转的 3d 圆环。圆环的明亮处，阴影用不同字符表示。通过不断刷新输出字符显示动态的圆环旋转。程序运行效果截图如下：



程序想法来源于 [1]

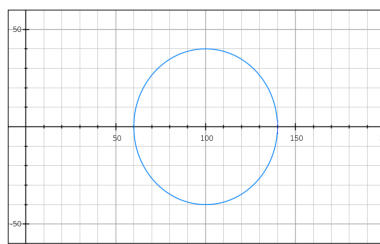
2 功能实现

整个程序分为：

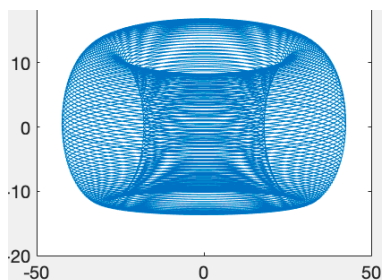
1. 从圆环的公式得到圆环上离散的点的坐标。
2. 将圆环上每一点映射到二维画布上。
3. 得到每一圆环上点的亮度。
4. 更新终端图片输出。
5. 翻转圆环，回到第 2 步重新将圆环映射到二维平面，得到亮度，刷新显示

2.1 圆环取点

1. 在 x-y 平面生成一个圆心在 (100, 0, 0)，半径为 40 的圆。



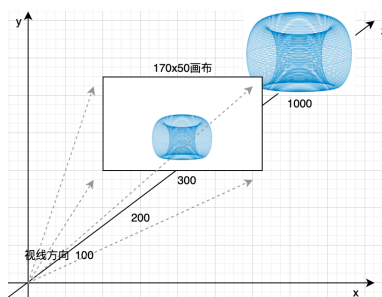
2. 将圆沿 y 轴旋转，生成一个中心在 $(0, 0, 0)$ 坐标原点的圆环，圆环截面即为 1. 中生成的圆。1. 中圆上的点划出的轨迹如下图：



3. 将圆环移动到中心位置为 $(0, 100, 1000)$ 处，此步是为了方便将原点作为视线出发点，将圆环映射到二维平面。

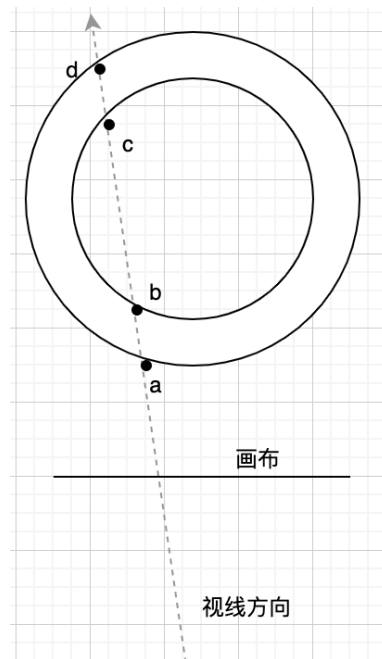
2.2 映射二维平面

程序将圆环投影到位于 $z = 300$ ，长 170 宽 50 的画布上。



投影使用透视投影^[2]，即坐标为 (x, y, z) 的点将被投影到 $(\frac{300x}{z}, \frac{300y}{z}, 300)$ 处。此时所有坐标都位于 $z = 300$ 的平面上，即位于画布上。

由于圆环为不透明的，部分像素是无法被看见的。为了过滤这些像素，不将它们显示在画布上，对于处于同一视线上的点总是取距离观测者最近的点，而忽略视线上剩下的所有点。如下俯视图中，abcd 都经过视线，但画布上将只有 a 的投影



2.3 得到像素亮度

对于圆环上每一像素，其反射的光线强度取决于其在圆环上的位置是否朝向光线。这里假设圆环发生的是漫反射，直接被光线照射的位置亮度最大，而不是像镜面反射那样将光线反射进视线最多的区域亮度最高^[2]

算法实现为：对于每一 2.2 中过滤后的所有像素，其在圆环上占的一小块区域可以看做一个平面。对于这些平面，得到垂直于平面的向量，和光线向量得到点乘结果。

点乘结果为正值，代表向量和光线在同一直线上，方向相同。可以判断像素是背光的像素，没有被光线照射，类似于俯视图中的 b d 点

点乘结果为 0，代表向量和光线垂直，则平面几乎没有被光线照射，亮度为 0

点乘结果为负值，代表向量和光线相逆，平面正对光线。则点乘结果绝对值越大，平面越被光线直射，亮度越大

得到亮度数值后，每一数值被对应到一个字符，使用的字符包含.,-;=!*#\$\$@，选取这些字符的依据是笔画多少。例如字符 @ 拥有较多笔画，则对应亮度较大的像素

2.4 旋转圆环

为了得到一个不断变化的圆环图像，程序每 0.1 秒对每一像素延 $y = 100, z = 1000$ 的直线翻转（即令圆环在原位置沿 x 轴翻转）。翻转后重新进入 2.2 将圆环映射进二维画布，得到亮度，输出到终端

2.5 更新终端输出

经过 2.4，程序已经得到一个长宽对应画布大小的矩阵。矩阵中每一个元素为一个字符代表像素亮度。为了在终端打印这个矩阵，并在经过 2.5 中翻转后更新终端中的矩阵，程序使用 python 包 `curses` 不断刷新输出。

3 Reference

- [1][https : //www.youtube.com/watch?v = sW9npZVpiMI](https://www.youtube.com/watch?v=sW9npZVpiMI) $t = 100s$
- [2][https : //materials.doc.ic.ac.uk/resources/2021/60005](https://materials.doc.ic.ac.uk/resources/2021/60005)
- [3][https : //docs.python.org/3/howto/curses.html](https://docs.python.org/3/howto/curses.html)