在圆中均匀分布随机点

计算机图形学

一.适用场合

在《光线跟踪算法技术》一书当中，有一个章节叫做《圆采样映射》。在通过圆形光源实施着色操作时，需要获取分布于单位圆上的采样点。实际上，这章提出的引申问题是，如何在一个圆内均匀分布点。在原书中，采用了一个比较复杂的方法去生成。这里介绍一种实现稍微简单点的方式。

二.计算流程

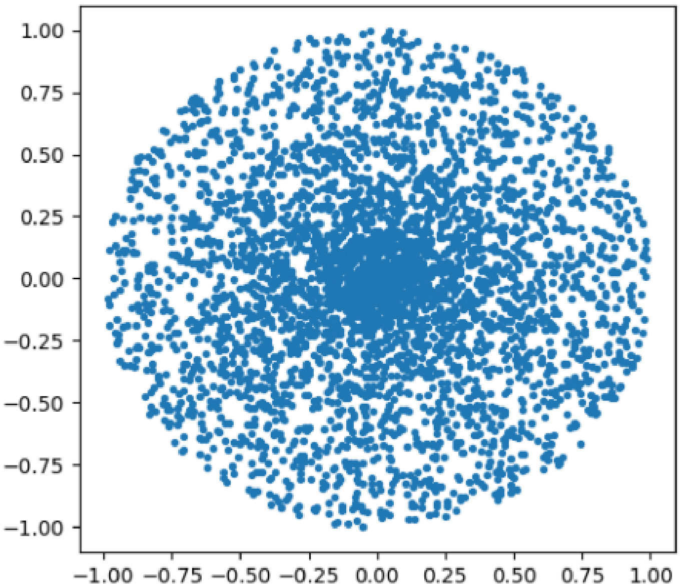
1.第一版算法

首先我们有一个均匀分布于[0,1)的随机数发生器，以及一个半径为R的圆，下面的问题是如何才能将[0,1)的随机数转化为圆内的点。
最直接想到的办法就是极坐标，为简化起见，假设圆心在原点，定义圆内的一个随机点P_0，就可以用(θ, r)来定义。随机变量θ是取值范围$[0, 2\pi]$的角度，随机变量r是取值范围[0,R]的值。
这样，第一版算法就出来了：

$$r = random() * R$$
$$\theta = random() * 2\pi$$

而点的(x,y)坐标为

$$x = \cos\theta * r$$
$$y = \sin\theta * r$$

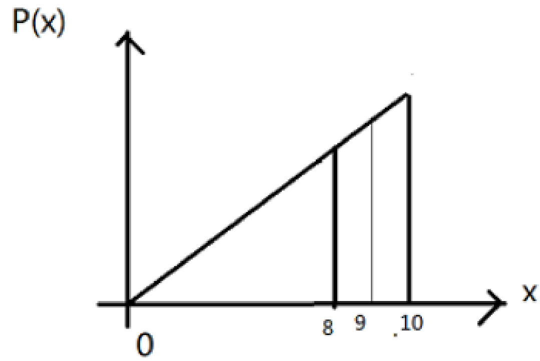


很可惜，这么做是不行的，很明显分布的点趋于中心。问题也很明显，如果要得到一个均匀分布的圆，r 的随机就不能是均匀的，而是越往边缘分布应该越密，越往中心越稀疏。但应该如何确定应该怎么分布呢？但首先的一个问题就是，就算我知道了它的分布密度，我该如何去实现生成这种分布密度的随机数的生成器呢？

2.反函数法

反函数法给出了由归一化[0,1]均匀分布的随机变量转化为符合特定分布随机变量的方法，流程是这样的：

1. 有一个概率密度函数f(x)，我们要求生成的随机变量服从它的分布。
2. 计算f(x)的原函数F(x)，这个原函数被叫做累积概率分布函数
3. 计算这个原函数的反函数$F^{-1}(x)$，当反函数的输入变量x属于[0,1]区间的时候，$F^{-1}(x)$的函数值符合概率密度函数f(x)的分布。当然这个方法只适用于f(x)的原函数F(x)求得出来，并且$F^{-1}(x)$也求得出来的时候才能用。
   对于反函数法，有一种简单的理解方式，f(x)是概率密度函数，假如其概率密度函数的图像是这样的：



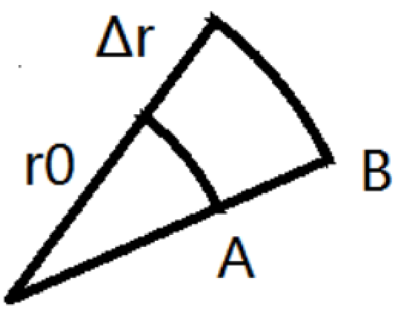
X取值[0,10]，并且由概率密度函数的要求，这个三角形总面积为1，即

$$\int_{-\infty}^{+\infty} f(x)dx = 1$$

在这里则简化为了$\int_0^{10} f(x)dx = 1$。
累积概率分布函数$Fk = \int_0^k f(x)dx$，这里指的是随机变量落到[0, k]区间的概率有多少。
因为F(x)实际上是一个映射，映射的是落到的区间–>事件发生的概率，那么反函数$F^{-1}(x)$说的就是给定了事件发生的概率–>它应当落到哪个区间里。
那么有一个随机变量x在[0,1]均匀分布，那么，经过$F^{-1}(x)$变换后，落在[9,10]区间内的概率是不是应该大一些呢？当然应该，这个区间内的面积占比就是比[8,9]更大一些。
因此很容易能理解，反函数法可以将均匀分布的[0,1]随机变量，转换为符合特定概率密度函数分布的随机变量。
根据这个步骤，就可以找到服从随机变量r的分布的随机数发生器。但要从头来，首先找到随机变量r的概率密度函数。

3.寻找概率密度函数表达式

圆面积的公式为$s = \pi * R^2$，假如随机点要在圆内均匀分布，按字面上理解，意思应该是每单位的面积点的数量要相同，即数量要和面积呈正比。那么，接下来的工作就是找到在半径R上的点的分布密度。



这是一个扇形，张角记为θ，由圆心到A的距离为r0，由A到B的距离为Δr，那么我想计算AB这段的面积，应该有如下公式计算得出

$S_0 = \frac{\pi*(r_0+\Delta r)^2*\theta}{2\pi} - \frac{\pi*r_0^2*\theta}{2\pi}$
$= \frac{\theta}{2} * (2 * r_0 * \Delta r + \Delta r^2)$

相应的，在R上取另一段长度记为r_1，也有对应在r_1处的面积。

$S_1 = \frac{\pi*(r_1+\Delta r)^2*\theta}{2\pi} - \frac{\pi*r_1^2*\theta}{2\pi}$
$= \frac{\theta}{2}*(2*r_1*\Delta r+\Delta r^2)$
下面计算二者的比值

$$\lim_{\Delta r->0}\frac{S_0}{S_1} = \frac{r0}{r1}$$

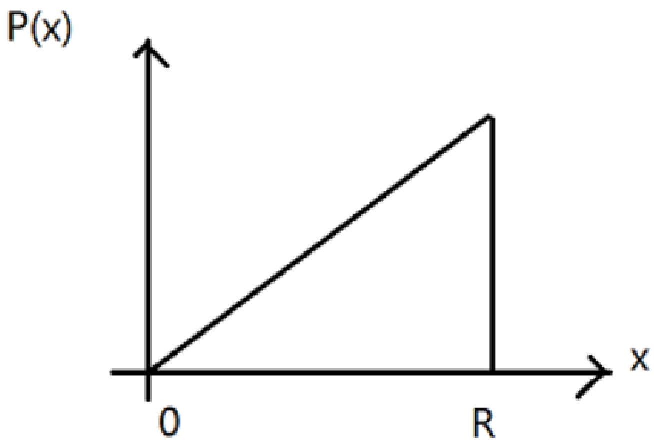
很明显位于r_0和r_1处的扇形委员面积正比于半径之比。
因此可以断定，点的密度应当正比于取的r。因此在半径R上的随机函数，其概率密度函数f应当满足于如下条件：

$$\frac{f(r_0)}{f(r_1)} = \frac{r_0}{r_1}$$

这里很好理解，既然点的数量正比于r，这也就是说其概率密度正比于r。
因此我们构造概率密度函数：

$$f(x) = k*x$$

其函数图像长这个样子：



其中，x的取值范围就为[0,R]。但k还不确定，所以要首先确定k的值。
根据概率密度函数的要求，其积分结果等于1。

$$\int_0^R kxdx = \frac{1}{2}kx^2\Big|_0^R = \frac{1}{2}kR^2 = 1$$

得出

$$k = \frac{2}{R^2}$$

由此，得到累积概率分布函数F(x)的完整形式：

$$F(x) = \int\frac{2}{R^2}xdx = \frac{2}{R^2}(\frac{1}{2}x^2+C)$$

当x=0时，累积分布F(x)应当为0，当x=R时，累积概率分布F(x)应当为1。由此得出C=0，即累积概率分布函数最终版本为

$$F(x) = \frac{1}{R^2}x^2$$

4.推导由[0,1]均匀分布转换为圆内点概率分布的公式

获得反函数：

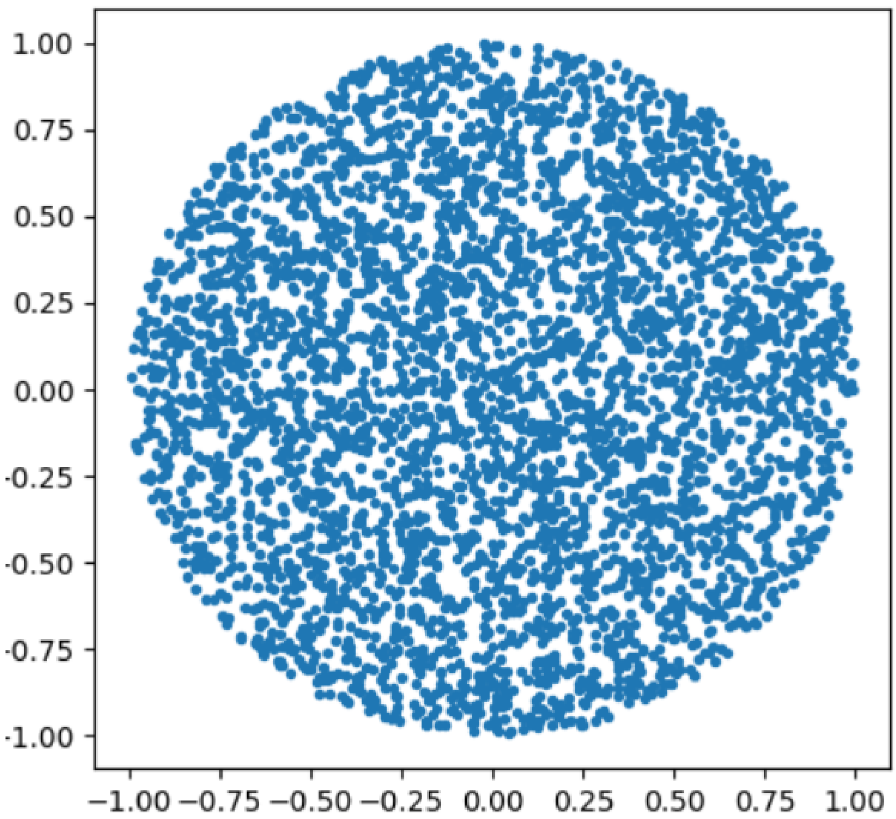
$$F^{-1}(x) = R * \sqrt{x}$$

根据反函数法，x为[0,1]均匀分布的随机数时，符合概率密度函数f(x)的分布。最终，我们就找到了这个随机数发生器。

5.代码实现

因此最终实现的代码为：

```
@staticmethod
    def GenCircleRandomPts(r, pt_cnt):
        x_list = []
        y_list = []
        for k in xrange(pt_cnt):
            #首先生成角度theta [0,2*pi]
            random_theta = random.random() * math.pi * 2
            #生成距离圆心的长度
            random_r = math.sqrt(random.random()) * r
            x = math.cos(random_theta) * random_r
            y = math.sin(random_theta) * random_r
            x_list.append(x)
            y_list.append(y)
        return [x_list, y_list]
```



最终结果也符合均匀分布的要求。