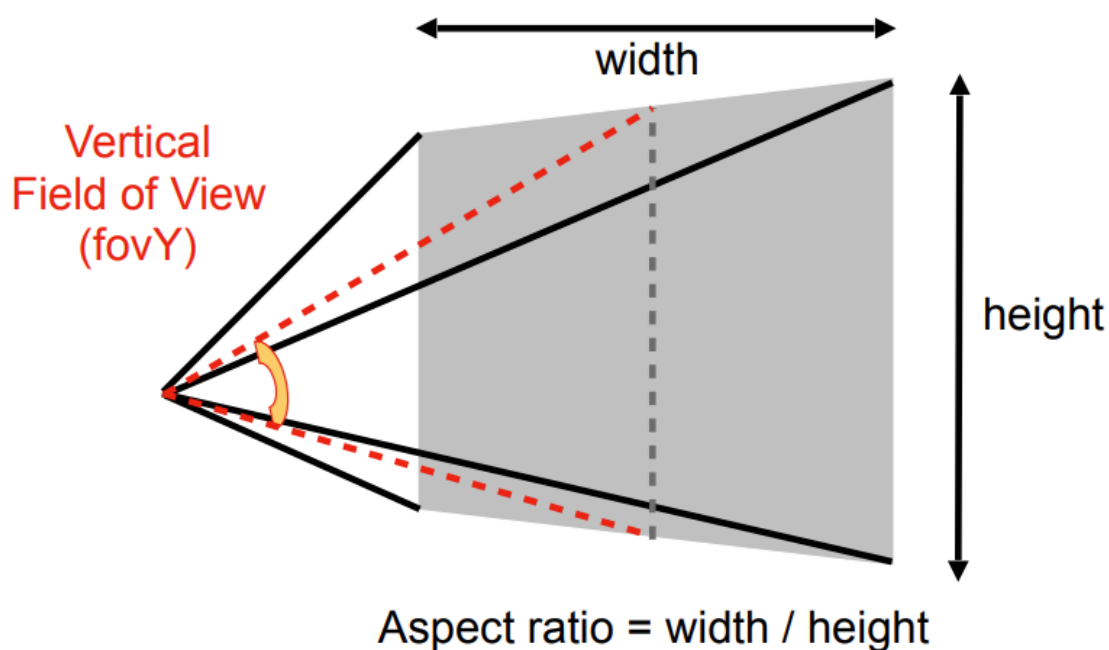


# Lecture 05

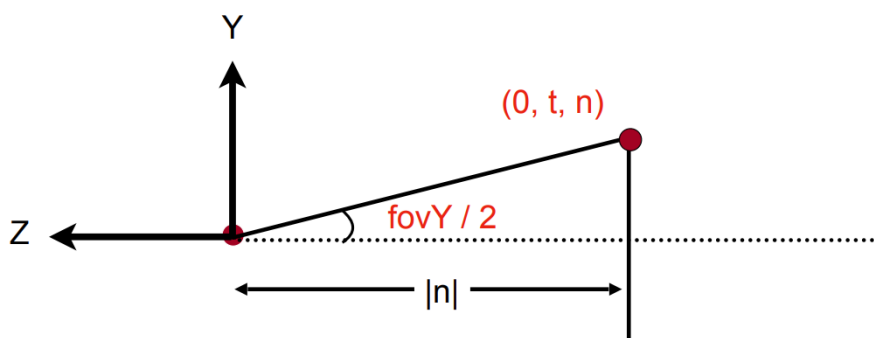
## 1. 透视投影

灰色平面代表近平面，由四个变量表示  $(l, r, b, t)$ ，先申明一些概念

- $l = -r, b = -t$
- Aspect ratio 宽高比
- Vertical Field of View(fovY) 垂直可视角度，即下图红线之间的角度



由已知变量可计算出 宽高比 和 fovY



$$\tan \frac{fovY}{2} = \frac{t}{|n|}$$

$$aspect = \frac{r}{t}$$

## 2.Canonical Cube to Screen

**MVP** 代表模型变换，视角变换，投影变换，在做到这个 [Lecture 04](#) 之后，我们需要把获得的 Cuboid 转化到屏幕上。

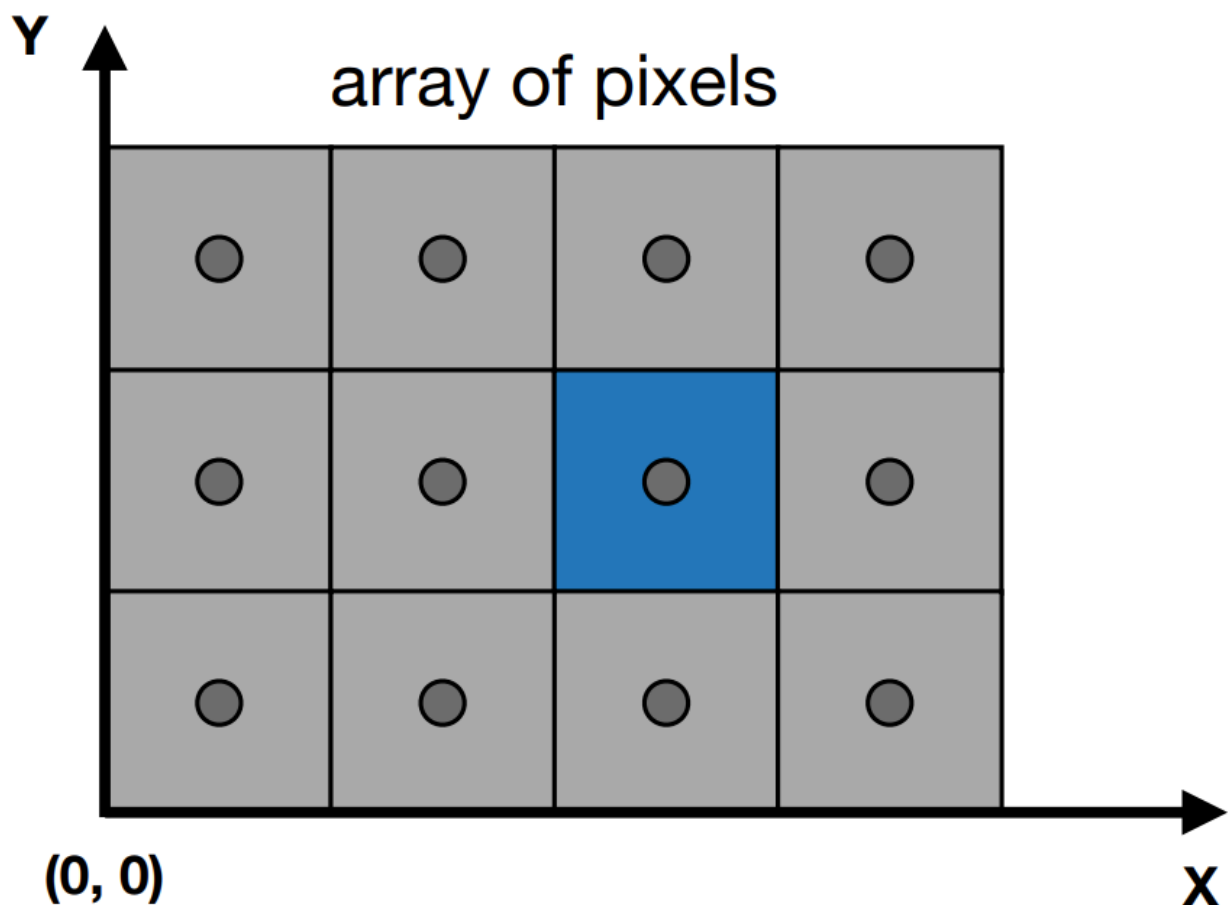
### 1.Screen

- Raster: 德语中的 Screen, 而 Rasterize 就代表 drawing onto the screen
- 像素: 一个小方块, 并且其内有统一的, 由红绿蓝三色组成的颜色。

### 2.屏幕空间

首先定义屏幕空间的一些性质

- 原点从  $(0,0)$  开始, 包含  $(width, height)$  的空间
- 每个像素用坐标  $(x, y)$  表示
- 像素的中心点在  $(x + 0.5, y + 0.5)$



再将 Cuboid 投影到屏幕上, 此时 **首先忽略 Z 轴**。并且需要完成两件事

- 将  $(X, Y)$  轴上  $(-1, 1)$  的范围变换成  $(width, height)$  的范围
  - 将图像原点移动至  $(0, 0)$
- 根据这两个步骤，我们可以得到 **视口变换矩阵 (Viewport transform)**

$$M_{viewport} = \begin{pmatrix} \frac{width}{2} & 0 & 0 & \frac{width}{2} \\ 0 & \frac{height}{2} & 0 & \frac{height}{2} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

## 3.Rasterization 光栅化

### 1.多边形

- **光栅化** 是将向量图形格式表示成图像转换成位图以用来显示器或打印机输出的过程
- Lecture 04 以及 Lecture 05 前面已经介绍了如何将空间中的物体变成屏幕上的一个多边形，而在具体的光栅化过程中，还可以将复杂的多边形由简单的多边形来表示。



图上的老虎就是由众多的多边形组成的

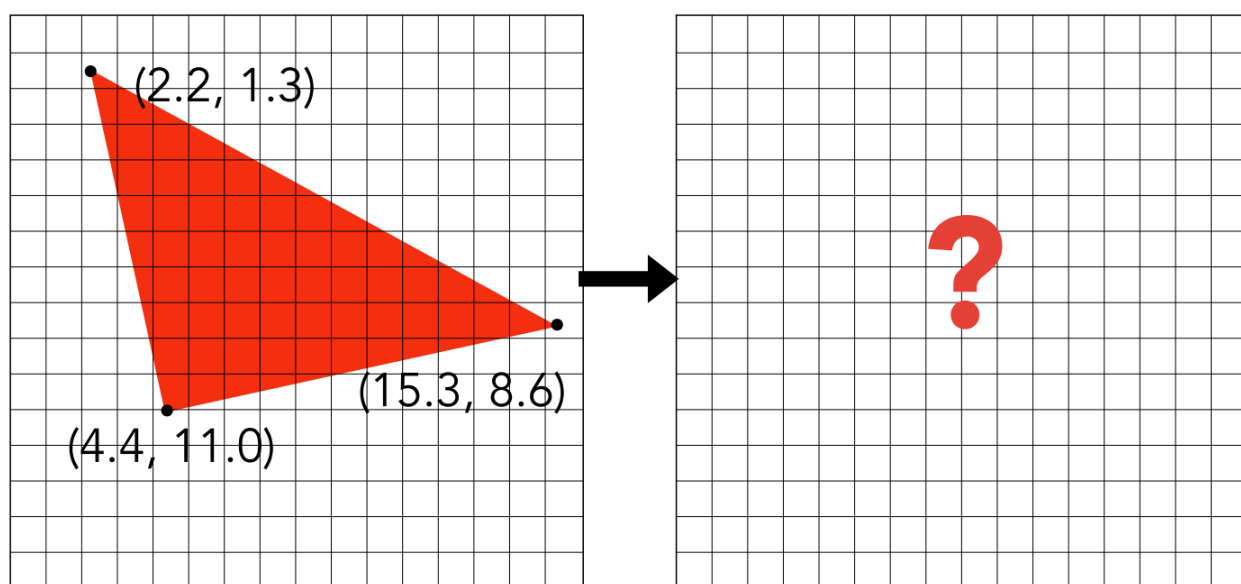
### 1.三角形

其作为多边形的基础，和最小的多边形，有以下的优势

- 任意一个三角形的所占区域均在 **同一平面**
- 易于判断点和三角形的 **内外关系**
- 易于用在 **重心插值(barycentric interpolation)** 和 **渐变** 上

## 2.采样

- 定义：利用像素的中心对屏幕空间进行采样，包含时间(1D)，区域(2D)，方向(2D)，体积(3D)多种采样
- 引入问题：下图的输入中，三角形在某些单位的像素中仅仅占了部分空间，而在绘制时需要判断这些点是否在三角形内，以便于稍后的着色等工作



Input: position of triangle vertices projected on screen

Output: set of pixel values approximating triangle

### 1.2D采样

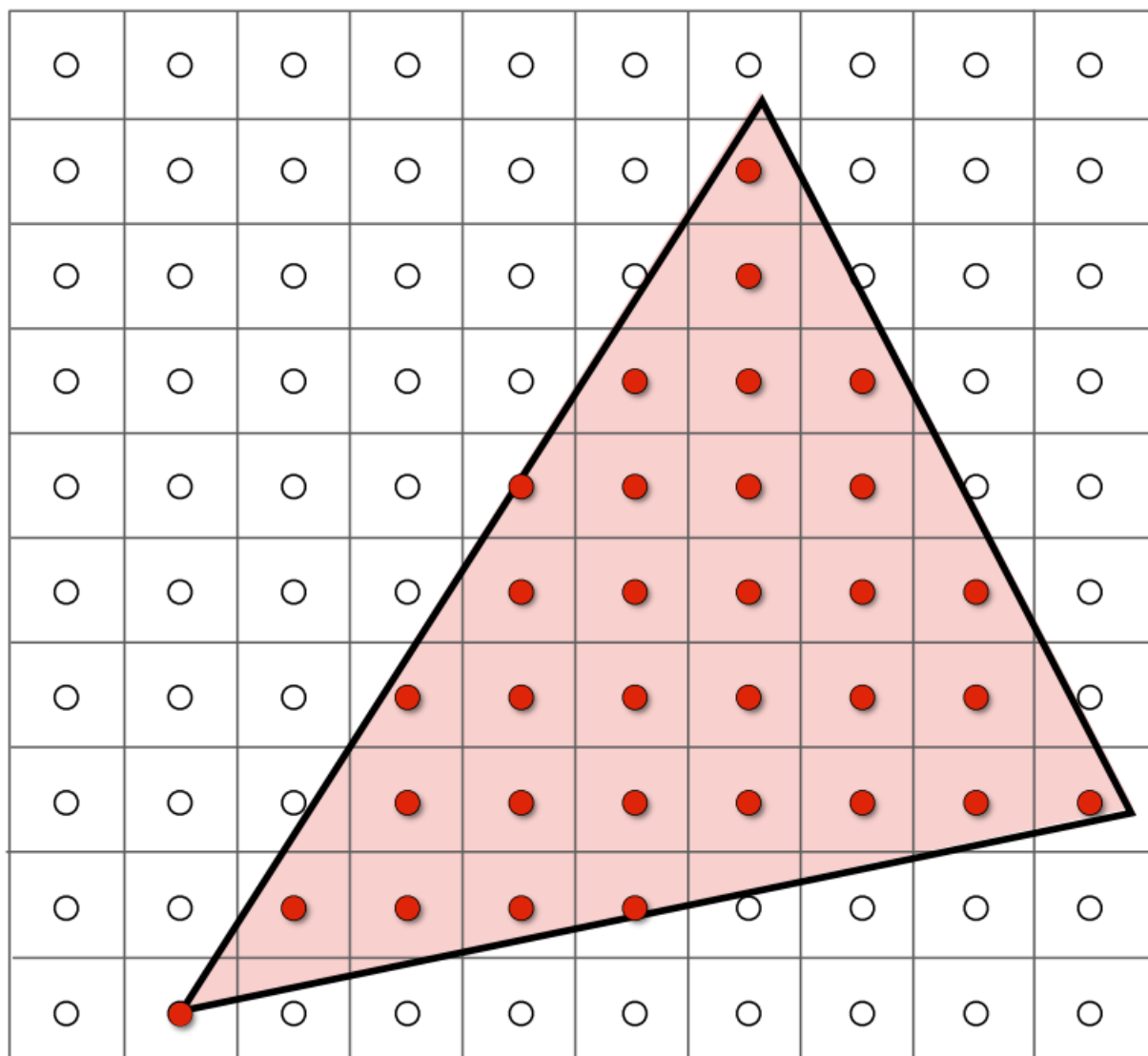
首先我们给出采样最基础的代码，即在 2D 空间中判断每个点是否应当采样

```

for(int x = 0; x < x_max; ++x) {
    for(int y = 0; y < y_max; ++y) {
        image[x][y] = inside(tri, x + 0.5, y +
0.5)
    }
}

```

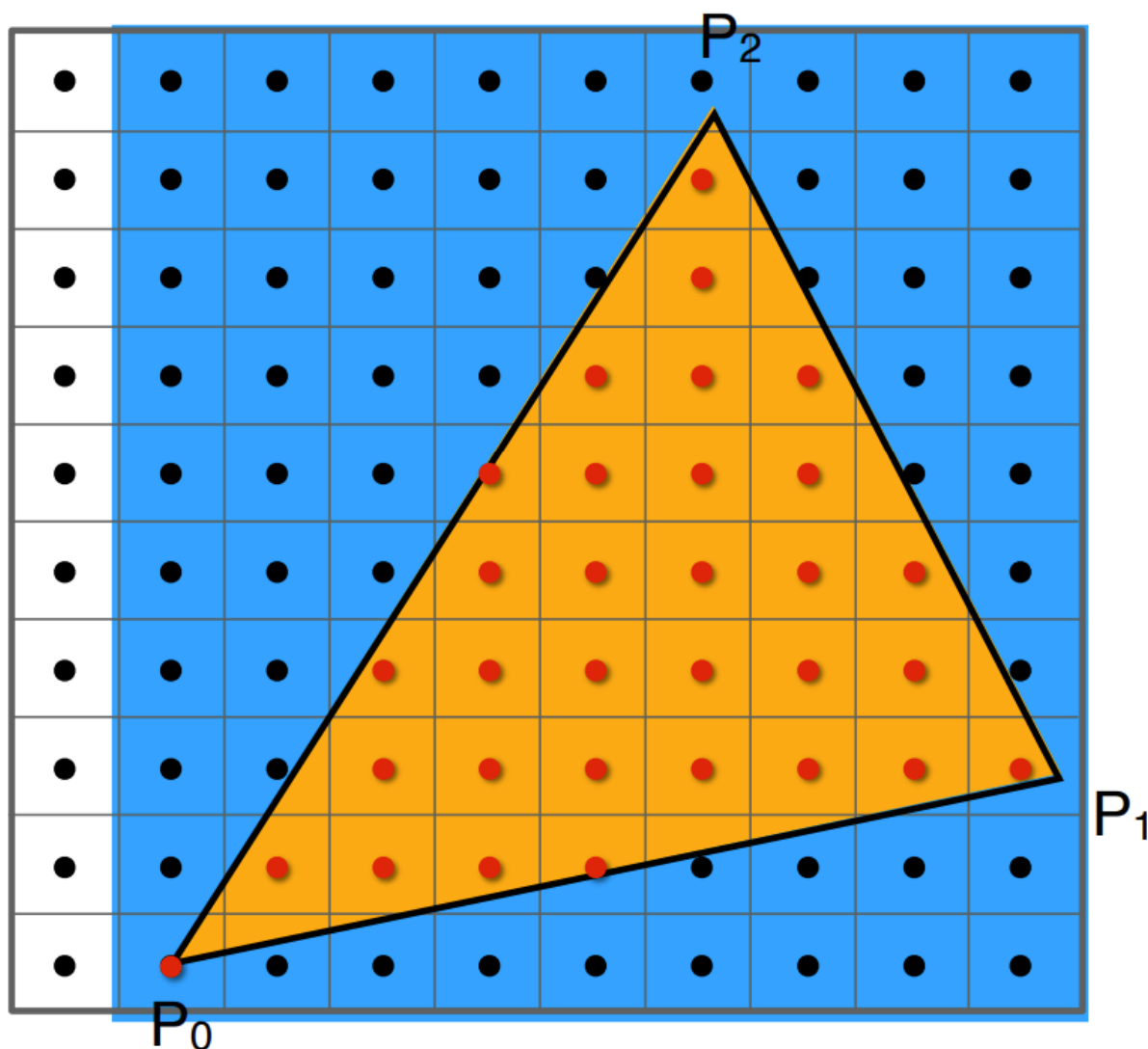
而最关键的是对于 *inside* 函数的实现 以及采样范围的选择



- 具体过程：判断每个像素中心是否在三角形内
- 采样细节：实现函数 *inside(tri, x, y)*，具体使用Lecture 02 的 叉乘性质判断点与三角形的关系
- 边界问题：采样点在三角形/多边形的边上，处理方式为 **不处理/特殊处理**，本节课不做处理。

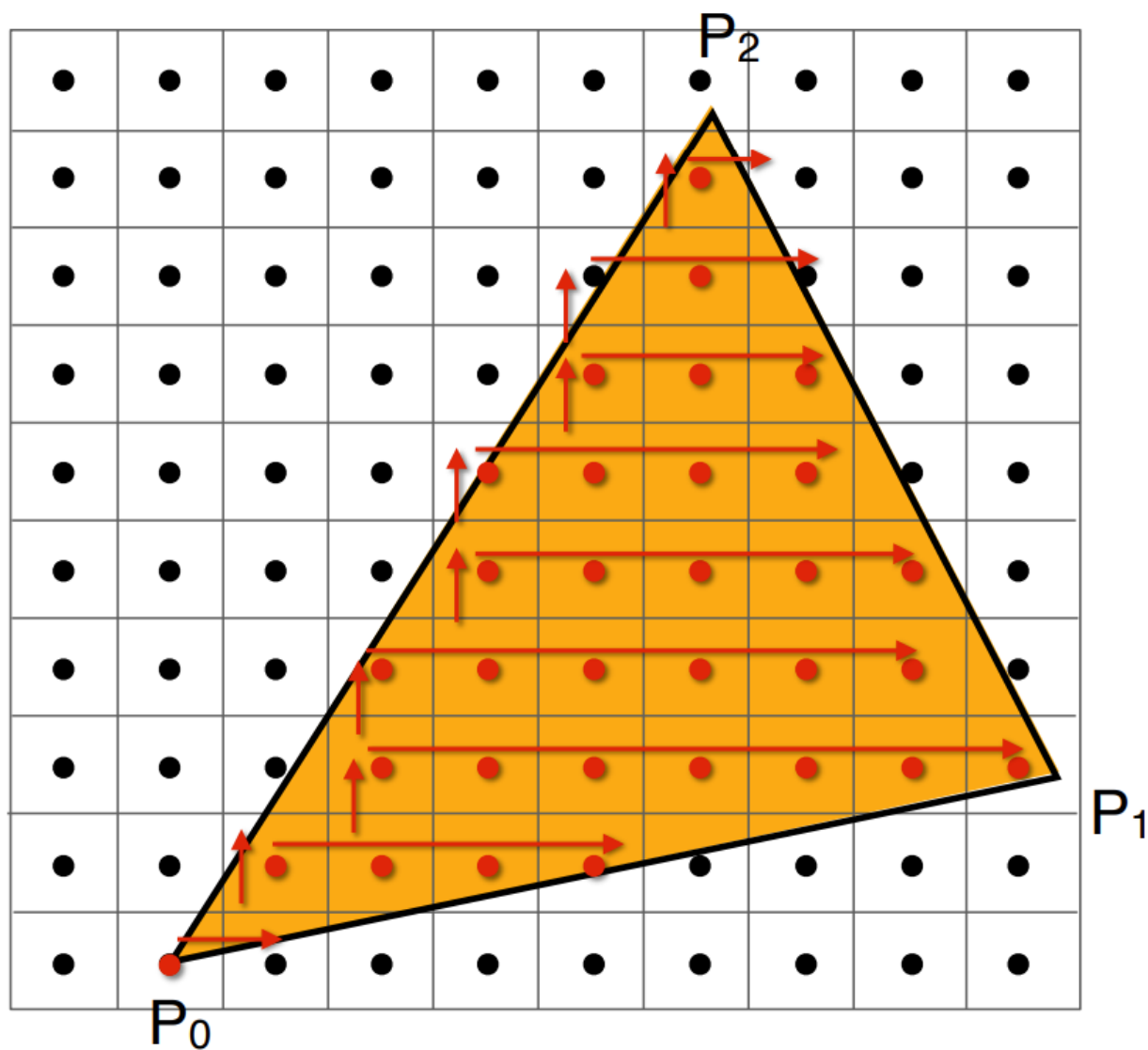
## 2.采样范围

**Bounding Box**, 即包围盒



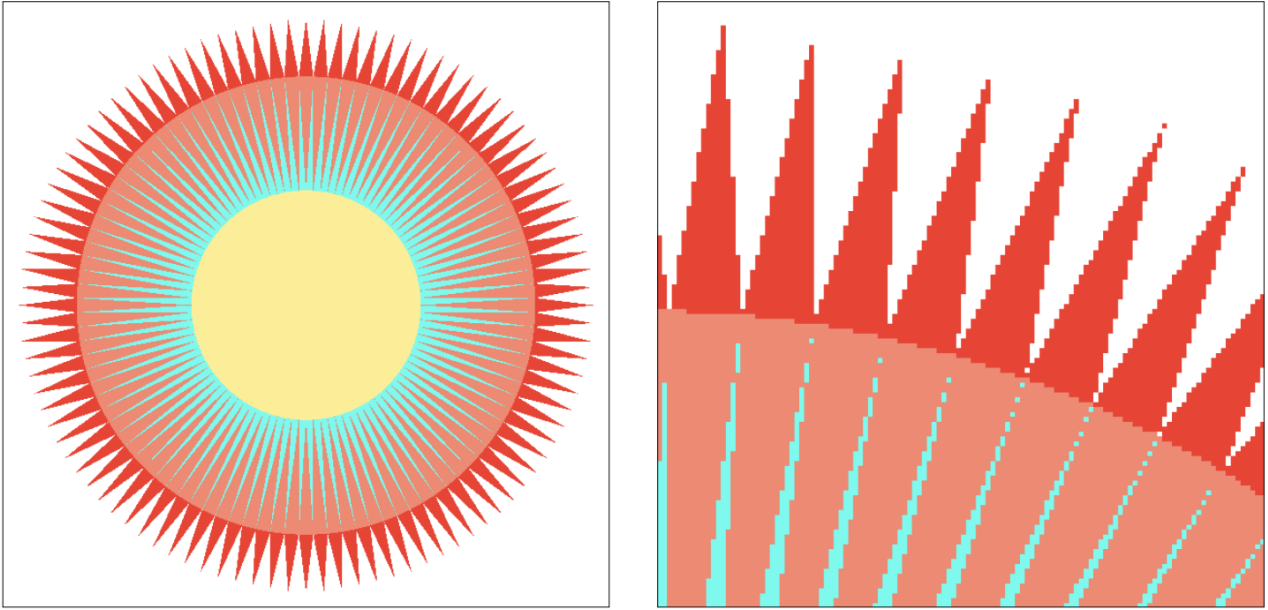
只有在蓝色矩形区域内的像素，即称为 Bounding Box内的像素，才是三角形 2D采样的像素

**Incremental Triangle Traversal**



针对三角形的每一行的左右范围取定每一行的采样范围，适用于扁长的三角形，可以比 Bounding Box 方法选定更少的像素，并且更快。

### 3.走样 Aliasing



由于采样率和像素本身大小的问题，会导致走样现象。