



# 计算机图形学小白入门

——从0开始实现OpenGL

几何顶点架构设计-VBO



授课：赵新政  
资深三维工程师

专注3D图形学技术  
教育品牌

## 顶点数据组织-VBO

- 对于一个三角形，我们可以把其不同的属性 (attribute) 组织成不同的数组内存空间(float\*)，让其更解耦
- 每个不同的内存空间，我们称之为：VertexBufferObject (VBO)

position

x0	y0	z0	x1	y1	z1	x2	y2	z2
----	----	----	----	----	----	----	----	----

color

r0	g0	b0	r1	g1	b1	r2	g2	b2
----	----	----	----	----	----	----	----	----

uv

u0	v0	u1	v1	u2	v2
----	----	----	----	----	----

## 顶点数据组织-VBO-Interleaved

- 对于一个三角形，我们可以把其不同的属性（attribute）组织成一个**数组内存空间**(float\*)，如图：

x0	y0	z0	r0	g0	b0	u0	v0	x1	y1	z1	r1	g1	b1	u1	v1	x2	y2	z2	r2	g2	b2	u2	v2
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

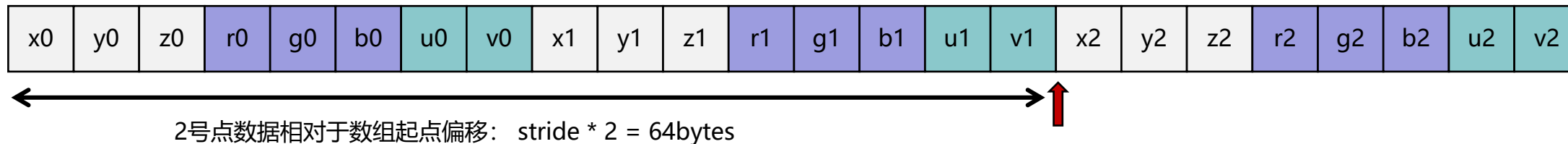
- 本数组也可以称为一个VBO，被送入管线进行渲染
- 此种数据排布方式，称为交叉数据缓存（Interleaved buffer）

## 顶点数据组织-VBO-描述

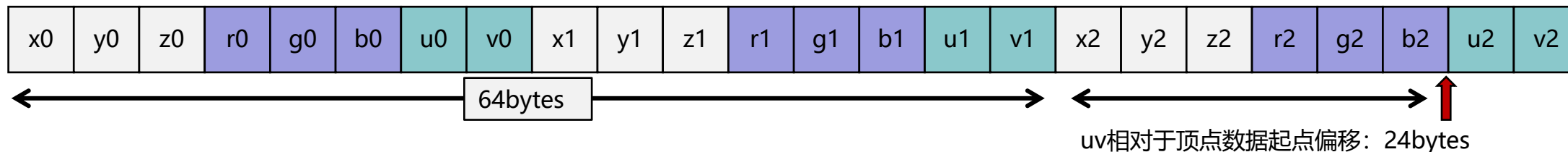
- 当GPU拿到一个VBO，显然有两种可能：某一个属性的vbo/一堆属性构成的vbo
- 对于某一个属性VBO，需要知道每个属性由**几个float构成**（position为3个，uv为2个）
- 对于interleaved buffer则需要知道：
  - 每个顶点属性读取的开始位置；
  - 某个顶点属性相对于开始位置偏移多少进行读取

## 顶点数据组织-VBO-描述-Interleaved Buffer

- 每个顶点数据长度为:  $8 * \text{sizeof}(\text{float}) = 32\text{bytes}$  -> **Stride=32**



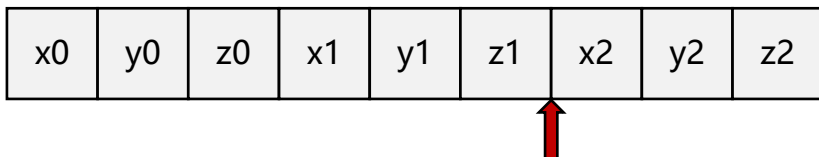
- uv属性相对于每个顶点数据内部偏移:  $3 * \text{sizeof}(\text{float}) + 3 * \text{sizeof}(\text{float}) = 24\text{bytes}$  -> **offset=24**



- uv由2个float类型数据构成, 即**itemSize = 2**
- 最终可以得到:  $p + \text{stride} * \text{index} + \text{offset}$ 的位置读取itemSize个float

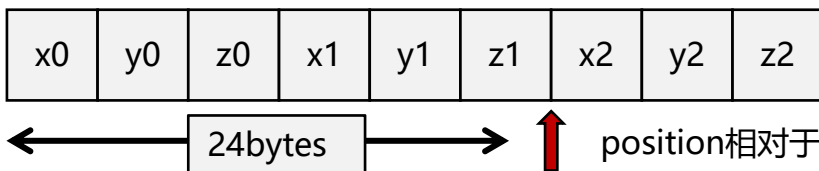
## 顶点数据组织-VBO-描述—Single Buffer

- 每个顶点数据长度为:  $3 * \text{sizeof}(\text{float}) = 12\text{bytes}$  -> **Stride=12**



2号点数据相对于数组起点偏移:  $\text{stride} * 2 = 24\text{bytes}$

- position属性相对于每个顶点数据内部偏移:  $0 * \text{sizeof}(\text{float}) = 0\text{bytes}$  -> **offset=0**



position相对于顶点数据起点偏移: 0bytes

- position由3个float类型数据构成, 即**itemSize = 3**
- 最终可以得到:  $p + \text{stride} * \text{index} + \text{offset}$ 的位置读取itemSize个float

## 顶点数据组织-VBO管理

- 每个VBO都会有自己的ID号，设计一个map结构，用于存储程序中分配出来的VBO及ID的键值对
- 当读取第i个顶点的某个属性时候，需要知道：
  - 从ID为几的VBO当中读取
  - stride/offset/itemSize

如何管理Description呢？

