坐标系对象介绍

概要：*在坐标系变换运算时大量应用矩阵运算，或者更一般的情况下使用张量运算。前者属于"元"数学对象，概念上不是为了坐标系变换设计，计算过程中存在冗余以及便捷性问题；后者难以掌握，而且比较难以数值化使用。本文介绍一种为专门为坐标系变换封装的结构对象来作为矩阵以及张量运算的替代。*

1. 背景

坐标系一直是最为抽象与困难的数学物理对象，发展出张量运算以后更加抽象，必须经过多年训练才勉强掌握。 所以尝试使用一种简化方法来处理好坐标系运算难题。坐标系变换常使用矩阵进行，但作为数学对象，矩阵不是为坐标系变化定制的，矩阵过于数学化，含义模糊。 张量过于抽象，不仅难以掌握，计算机也难以量化，需要专门为坐标系变换设计的概念。

2. 设计

设计一个数学对象名为 coord，基于群论思路围绕它设计四则运算。

a) 坐标系结构体的定义：

- 三维空间中的坐标系由一个原点加上三个方向轴和三个缩放分量组成，

  分别对应**位移、旋转、缩放**三种变换(C++版本):

```
struct coord
{
    vec3 ux, uy, uz;        // 三个单位基向量
    vec3 scale;             // 缩放
    vec3 o;                 // 原点位置
}
```

  构造一个坐标系对象(C++版本)

通过三个轴

coord C(vec3 ux, vec3 uy, vec3 uz)

coord C(vec3 ux, vec3 uy);

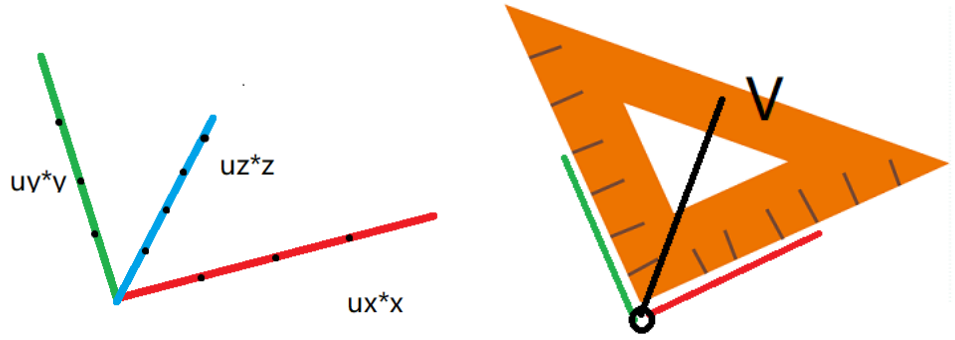
通过欧拉角

coord C(float angle, vec3 axis);

coord C(float pitch, float yaw, float roll);

（原点跟缩放可以直接设置）

b) 乘法：在某坐标系下定义向量



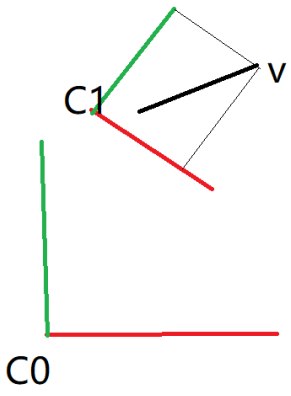
使用乘法实现向量从本坐标系变换到父坐标系下以及坐标系的合并运算

例如：V0 定义在世界坐标系 C0 下，V1 定义在 C1 坐标系

V0 = V1 * C1

C1,C2,C3 分别是三个场景节点下的本地坐标系，父->子方向：

V0 = V3 * C3 * C2 * C1 = V3 * (C3 * C2 * C1)



与向量交换后乘法运算:

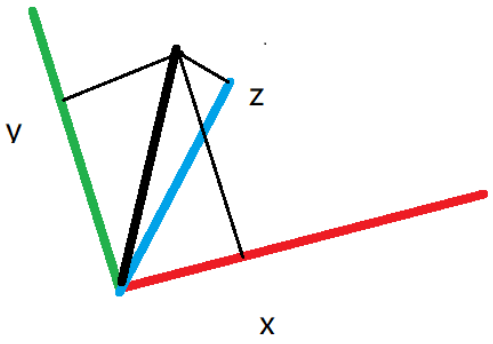
V * C != C * V

定义坐标系与向量相乘:

C * V = Lerp(ONE, C, V.dot(UX))

其中 Lerp 为线性插值函数，ONE = coord.ONE, UX = vec3.UX

c) 除法：用某坐标系量度一个向量



使用除法从父坐标系投影到本坐标系，例如:

V0 定义在世界坐标系 C0 下, V1 定义在 C1 坐标系

V1 = V0 / C1, V0 = V1 * C1

V2 定义在 C2 坐标系下则:

V2 = V0 / C2 = V1 * C1 / C2

C1, C2, C3 分别是三个场景节点下的本地坐标系，父->子方向，

V3 = V0 / C1 / C2 / C3 = V0 / (C1 * C2 * C3)

d) 通常的使用场景

1. 比如一个世界空间下的向量 Vw，转化到本地坐标系 C 下

   VL = Vw / C

   反之: Vw = VL * C

2. 世界坐标系与局部坐标系相互转化

   C = C3 * C2 * C1

   Vw = VL* C, VL = Vw / C

3. 多节点层级下使用

   节点 5 坐标系下定义向量 V5，在节点 2 层次的父节点下:

   V2 = V5 * C5 * C4 * C3

   其中每个坐标系都是本地坐标系

   反之: V3 = V2 / C3 / C4 / C5

4. 平级坐标系之间互换

   C0{ // C0 父坐标系下

   C1, C2; // 两个平级子坐标系

   }

   向量从 C1 下转化为 C2 坐标系下

   V2 = V1 * C1 / C2

5. 更多的运算

标量乘法：

C*k = C.scale * k

k 为浮点数字

四元数乘法

C*q = C.ux(q.angle, q.axis);

C.uy(q.angle, q.axis);

C.uz(q.angle, q.axis);

q 为四元数

加减法

C1+o = C1.o + o, C1-o = C1.o - o

o 为三维向量(vec3)

e) 加减法运算

在计算两个向量之差时：

DV = V1 – V2

设 V1 = v * C1, V2 = v * C2

DV = v * (C1 – C2)

这里定义一个加减法运算用来进行向量基本加减法：

V1 + V2 = v * (C1+ C2)

V1 - V2 = v * (C1- C2)

3. 在微分几何领域的应用

a) 向量的移动

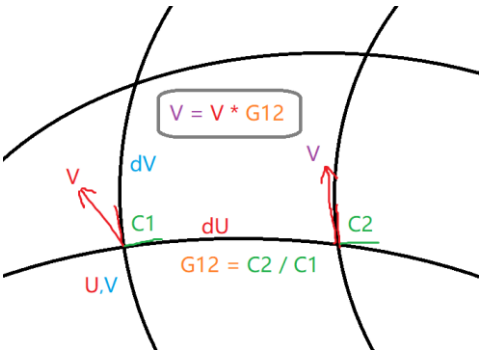
定义一个自然坐标系(假设它是平直空间，向量可以随意移动而不变)下 V,在弯曲空间坐标系下观察 V，不同点上 V 是不同的，故而坐标系跟位置有关，取相邻两点(1), (2)点处有向量 V1, V2，对应坐标系 C1, C2，那么：

V = V1 * C1 = V2 * C2 =>

V2 = V1 * C1 / C2，令 G12 = C1 / C2 =>

V2 = V1 * G12

在任意微分向量方向上移动：



任意微分向量：dW = Cw * dU

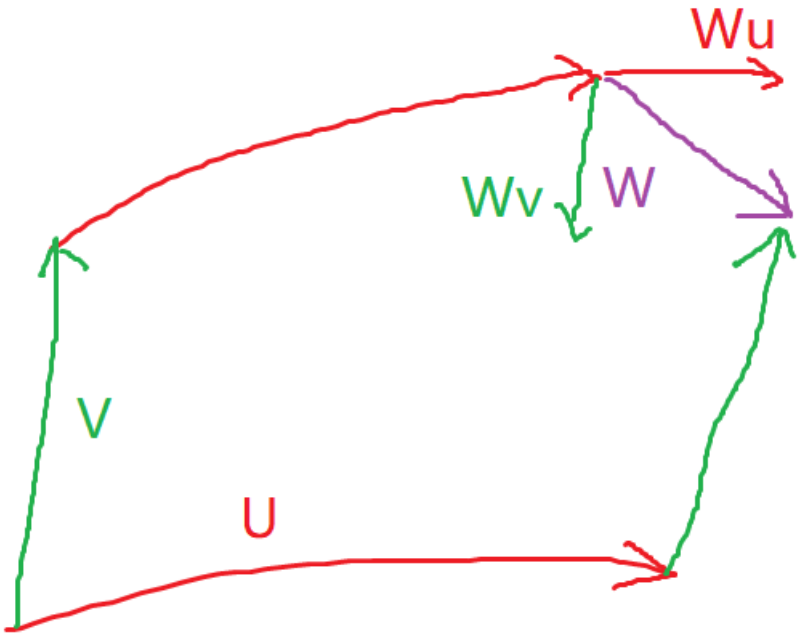
所以在任意微分向量方向上移动：

V2 = V1 * Gw = V1 * Gu * Cw

其中：Gu = G12 = C1 / C2

单位微分向量 dU 方向上的两个坐标系对象之商。

b) 曲率计算



在弯曲坐标系下，自然坐标系 x,y 轴的平行线投影得到的 u,v 曲线上 G12 分别在两个方向上对应 Gu,Gv，从(u1,v1)到(u2,v2) 计算两个路径的差别再加上修正项可得曲率公式为：

Ruv = Gu*Gv - Gv*Gu*Gu*Wu*Gv*Wv

其中：W = Wu + Wv = Gu * Wu + Gv * Wv

4. 与李群，李代数的结合

旋转矩阵 R 是李群的一个元素，我们的坐标系的乘法运算跟旋转矩阵是等价的，所以我们的坐标系 C 也是李群的一个元素，乘法为运算，0 元素为 ONE:

C∈{ONE, (*)}

向量叉乘对应的坐标系

v1 X v2 = v*C1 X (v * C2)

令 v = vec3.ONE 即 vec3(1,0,0)，那么 C * v = C，所以上式可以写为：

v1 X v2 = v*(C1 X C2)

我们定义一个坐标系叉乘运算,用李代数括号表达：

[C1, C2] = C1 * C2 – C2 * C1;

5. 结论

坐标系对象可以简化局部空间下的线性运算，在计算向量移动与旋转操作时可以适当设计一些运算，按照群论的规则形成统一的形式。可以应用于约束计算等需要大量线性运算的情况。通过坐标系对象把向量，矩阵，张量结合在一起，把李群跟李代数对象统一成一个环，形成了统一而连贯的形式。