Barycentric coordinates & Mesh transfer

重心坐标（Barycentric coordinate），即通过使用单纯形（Simplex）对一个点的坐标表示。比如，在二维平面内，使用一个三角形表示，在三维空间中，用一个四面体（Tetrahedron）表示。重心坐标可以理解为当该点坐标的数值为该单纯形的顶点质量时，该点即为该单纯形的重心。重心坐标分量可以为负数，而当且仅当所有分量为正的时候，点位于单纯形内部。

在计算机图形学中，重心坐标使用三个scaler描述任意一个落在三角形内的点。计算这样的坐标可以使用下面的等式：

$$P = wA + uB + vC,\ w + u + v = 1 \tag{1}$$

因此，重心坐标的一个相当重要的应用便是在三角形内插值顶点数据，比如法向、颜色、纹理坐标等等。下面通过射线检测的简单代码给出示例：

```
size_t face_idx = 0;
Vertex vt1 = vertices[faces[face_idx][0]];
Vertex vt2 = vertices[faces[face_idx][1]];
Vertex vt3 = vertices[faces[face_idx][2]];
if (rayTriangleIntersect(...)) // if intersects with this triangle
{
    Color color = w * vt1.color + u * vt2.color + v * vt3.color;
    Vector3 normal = w * vt1.normal + u * vt2.normal + v * vt3.normal;
    Vector3 reflect = ray.reflect(normal);
    // ...
}
```

备注

三角形内点的坐标也可以用UV坐标表示，如下：

$$\begin{gathered} P = A + u\overrightarrow{AB} + v\overrightarrow{AC} \\ = A + u(B - A) + v(C - A) = (1 - u - v)A + uB + vC \end{gathered} \tag{2}$$

因此，UV坐标实际上也是重心坐标的一种变种，也因此，重心坐标的三个参数我们写作 w, u, v 的顺序。

前置内容

仿射空间

仿射空间（Affine space）是点（points）和向量（vectors）的集合，在仿射空间中，点和点相减得到向量，点和向量相加得到另一个点，但是点和点不能相加。在仿射空间中没有原点（origin）的概念，向量也不和点唯一对应。我们可以描述一个仿射空间为：一个集合 A 、一个向量空间 $\overrightarrow{A}$ 以及一个映射关系 $A \times \overrightarrow{A} \to A$ 。该映射具有结合性（associative）、传递性（transitive），以及一一对应性（bijective）。

仿射组合

仿射组合（Affine combination）是特殊的线性组合，对于一个线性组合：

$$\sum_{i=1}^{n} \alpha_i \cdot x_i = \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_n x_n \tag{3}$$

当有：

$$\sum_{i=1}^{n} \alpha_i = 1 \tag{4}$$

时，为仿射组合。

对一个仿射组合的仿射变换等同于对其向量的仿射变换：

$$T \sum_{i=1}^{n} \alpha_i \cdot x_i = \sum_{i=1}^{n} T\alpha_i \cdot x_i \tag{5}$$

仿射变换

仿射变换（Affine Transformation）是对一个仿射空间进行一次线性变换，加一次平易变换。仿射变换保留了以下特征：

1. 点之间的共线性
2. 直线的平行性
3. 集合的凸性
4. 平行线段长度的比例
5. 不同质量的点组成集合的质心

仿射变换是仿射空间的一个自同构（Automorphism），即，仿射变换将一个仿射空间映射（可逆）回自身，并且保留其所有仿射子空间的维度。仿射子空间（Affine subspace）的定义为，对于一个仿射空间A，其子空间B的向量空间为A的向量空间的一个子集。

仿射变换可以描述为：

$$f(x) = A\vec{x} + \vec{b} \tag{6}$$

或者使用增广矩阵：

$$\begin{bmatrix} f(x) \\ 1 \end{bmatrix} = \left[\begin{array}{c|c} A & \vec{b} \\ 0 \cdots 0 & 1 \end{array}\right] \begin{bmatrix} x \\ 1 \end{bmatrix} \tag{7}$$

仿射相关

在n维空间中，n个线性相关（Linear dependent）的点（向量）被描述为：

$$\exists a_i \in \mathbb{R}, \vec{x}_i \in \mathbb{R}^n, i = 1, 2, \ldots, n \Rightarrow \sum_{i=1}^{n} a_i \vec{x}_i = \vec{0} \tag{8}$$

而如果不存在这样一组系数使得n个向量的组合为0，则这一组向量线性无关。一个空间的维度由该空间中最大数量的线性无关向量的个数决定。

仿射相关（Affine dependent）则是在线性相关的基础上，增加一个条件，即 $\sum_{i=1}^{n} a_i = 1$ 。或者，也可以描述为，对n个点，如果其中任意一个点不能被其他点的仿射组合表示，则该n个点线性无关；相反，则线性相关。

由于仿射空间没有原点的概念，我们可以从n个点选择一个作为原点，如果剩下的n-1个点相对于原点线性相关，则该n个点仿射相关。

在三维空间中，三个点如果共线或者共点，则为仿射相关；否则，为仿射无关，其仿射组合张成一个二维平面。四个仿射无关的点通过仿射组合张成一个三维空间。

重心坐标

在欧氏空间或者n维仿射空间A中有n+1个仿射无关的点 $A_0, ..., A_n$ ，任给 $P \in A$ ，有一组系数 $a_0, ..., a_n$ 对任意O（原点）使下式成立：

$$(a_0 + ... + a_n)\overrightarrow{OP} = a_0\overrightarrow{OA_0} + ... + a_n\overrightarrow{OA_n} \tag{9}$$

则，由n+1个元素组成的坐标 $(a_0 : ... : a_n)$ 即为点P的重心坐标，具有以下特性：

1. 该坐标与原点的选择无关
2. 该坐标为齐次坐标，即该坐标在数值上的缩放并不改变这个坐标，某点P的重心坐标不唯一
3. 重心坐标具有仿射不变性

计算重心坐标

重心坐标的数值与其内部点P划分出的三个小三角形的面积有关，我们设有三角形 $\triangle ABC$ ，其内部有一点P，则有：

$$\begin{aligned} w &= \frac{\triangle BCP_{Area}}{\triangle ABC_{Area}} \\ u &= \frac{\triangle ACP_{Area}}{\triangle ABC_{Area}} \\ v &= \frac{\triangle ABP_{Area}}{\triangle ABC_{Area}} \end{aligned} \tag{10}$$

首先，我们熟悉一下三角形的面积计算公式：

$$\begin{aligned} \triangle ABC_{Area} &= \frac{||(B-A)|| \cdot ||(C-A)|| sin(\theta)}{2} \\ &= \frac{||(B-A) \times (C-A)||}{2} \end{aligned} \tag{11}$$

这里，θ 是AB与AC形成的夹角，根据上式，三角形的面积可以视作任意两边向量的叉乘的大小除以2，也就是三角形的未归一化的法向量的长度除以2。

```
Vector3 v01 = v1 - v0;
Vector3 v02 = v2 - v0;
Vector3 normal = v01.cross(v02);
float triangle_area = normal.length();
```

所以，重心坐标的uv可以由下式计算：

$$
\begin{aligned}
u &= \frac{\triangle ACP_{Area}}{\triangle ABC_{Area}} = \frac{||\overrightarrow{AC}\times\overrightarrow{AP}||}{||\overrightarrow{AB}\times\overrightarrow{AC}||} \\
v &= \frac{\triangle ABP_{Area}}{\triangle ABC_{Area}} = \frac{||\overrightarrow{AB}\times\overrightarrow{AP}||}{||\overrightarrow{AB}\times\overrightarrow{AC}||}
\end{aligned} \tag{12}
$$

这里，因为 $\overrightarrow{AB}\times\overrightarrow{AC}=\overrightarrow{N}$ 而且，由于 $\overrightarrow{AC}\times\overrightarrow{AP}$ 与 $\overrightarrow{N}$ 是同向的（P点在ABC内），为简化计算（避免开销较大的大小计算）我们有：

$$
\begin{aligned}
u &= \frac{||\overrightarrow{AC}\times\overrightarrow{AP}||}{||\overrightarrow{AB}\times\overrightarrow{AC}||} = \frac{(\overrightarrow{AC}\times\overrightarrow{AP})\cdot\overrightarrow{N}}{\overrightarrow{N}\cdot\overrightarrow{N}} \\
v &= \frac{||\overrightarrow{AB}\times\overrightarrow{AP}||}{||\overrightarrow{AB}\times\overrightarrow{AC}||} = \frac{(\overrightarrow{AB}\times\overrightarrow{AP})\cdot\overrightarrow{N}}{\overrightarrow{N}\cdot\overrightarrow{N}}
\end{aligned} \tag{13}
$$

此外，由于 $(\overrightarrow{AC}\times\overrightarrow{AP})\cdot\overrightarrow{N}$ 可用于判断P点是否位于三角形ABC内部，因此在ray tracing的管线中，实际上该值已经计算过，因此更进一步节省了运算开销。对于Ray Tracing中的射线求交，在之后的笔记中会谈到。

简化运算

对于计算机而言，两个三维向量的叉乘开销较大，下面用较为简单的算法计算：

$$
\begin{gathered}
P = wA + uB + vC = (1-u-v)A + uB + vC = A + u(B-A) + v(C-A) \\
\Rightarrow P - A = u(B-A) + v(C-A) \\
\Rightarrow \overrightarrow{AP} = u\overrightarrow{AB} + v\overrightarrow{AC}
\end{gathered} \tag{14}
$$

令 $\overrightarrow{AB}=\vec{v}_0$ ，$\overrightarrow{AC}=\vec{v}_1$ 以及 $\overrightarrow{AP}=\vec{v}_2$，并且 $d_{ij}=\vec{v}_i\cdot\vec{v}_j$ ，则有：

$$
\vec{v}_2 = u\vec{v}_0 + v\vec{v}_1 \tag{15}
$$

（15）两边点乘 v_0 ：

$$
\vec{v}_2\cdot\vec{v}_0 = u\vec{v}_0\cdot\vec{v}_0 + v\vec{v}_1\cdot\vec{v}_0 \tag{16}
$$

（15）两边点乘 $\vec{v}_1$ ：

$$
\vec{v}_2\cdot\vec{v}_1 = u\vec{v}_0\cdot\vec{v}_1 + v\vec{v}_1\cdot\vec{v}_1 \tag{17}
$$

得：

$$
\begin{aligned}
d_{02} &= d_{00}u + d_{01}w \\
d_{12} &= d_{01}u + d_{11}w
\end{aligned} \tag{18}
$$

解得：

$$
v = \frac{d_{11}d_{02} - d_{01}d_{12}}{d_{00}d_{11} - d_{01}^2} \qquad w = \frac{d_{00}d_{12} - d_{01}d_{02}}{d_{00}d_{11} - d_{01}^2} \tag{19}
$$

Mesh Transfer : SMPL to SMPLX

smpl是一个人体参数化模型，而smplx则是其升级版，包括带关节的手部模型以及带关节和表情的头部模型。smplx和smpl有着略微区别的pose和shape参数，因此，它们之间的参数并不是可互换的。因此，如果需要将smpl升级为smplx，先将smplx的网格拟合到目标smpl模型的网格上，然后再拟合smplx的参数，得到我们需要的结果。

具体做法如下：

首先，使用rest shape的smpl和smplx模型做registration，对每一个smplx模型上的顶点，在smpl模型上找到距离其最近的三角面，并且将smplx的该顶点用重心坐标表示在smpl模型的该三角面上。因此，我们得到以下两个数据：

1. 一个smplx顶点到smpl三角面的对应表
2. 一个smplx顶点在对应的smpl三角面上的重心坐标

同时，使用一个遮罩排除掉smplx上相对于smpl特殊的结构，比如眼球和口腔等等。

然后，在目标的smpl模型上，令 $f_i \in \mathbb{N}^3$ 为SMPLX第i个顶点对应的SMPL的三角面的三个顶点序号，$[a_i, b_i, c_i]$ 为SMPLX第i个顶点在该三角面中的重心坐标。则SMPLX的第i个顶点可以通过下面的计算获得：

$$v_i^{SMPL-X} = a_i * v_{f_i^0}^{SMPL} + b_i * v_{f_i^1}^{SMPL} + c_i * v_{f_i^2}^{SMPL} \tag{20}$$

这样，我们就获得了在SMPLX拓扑下的模型，然后使用优化拟合的方式，最终得到我们需要的SMPLX参数化模型的beta、pose等参数。

参考

[1] https://en.wikipedia.org/wiki/Barycentric_coordinate_system

[2] https://en.wikipedia.org/wiki/Affine_space

[3] https://www.scratchapixel.com/lessons/3d-basic-rendering/ray-tracing-rendering-a-triangle/barycentric-coordinates

[4] https://github.com/vchoutas/smplx/blob/master/transfer_model/docs/transfer.md