先考虑n维欧式空间（我们要做的就是$n=4$维欧式空间中的$m=n-1=3$维流形，设三维流形的空间为V)，取一组正交基底$\boldsymbol{\delta}=\{\boldsymbol{\delta}_1,\ldots,\boldsymbol{\delta}_n\}$。设$n-1$维流形在$n$维欧式空间的参数式$\boldsymbol{f}(\boldsymbol{u}),\boldsymbol{f}:\mathbb{R}^{n-1}\to\mathbb{R}^n,\boldsymbol{u}\in\mathbb{R}^{n-1}$为曲纹坐标。光线在流形参数方程$\boldsymbol{v}(t),t\in\mathbb{R},\ \boldsymbol{v}\in\mathbb{R}^{n-1}$。

在这个曲纹坐标系下的自然坐标架$\boldsymbol{r}=\{\boldsymbol{r}_1,\ldots,\boldsymbol{r}_m\}$为

$$\boldsymbol{r}_i=\sum_{j=1}^{n}\frac{\partial f^j}{\partial u_i}\boldsymbol{\delta}_j,\qquad \boldsymbol{r}=\boldsymbol{\delta}\boldsymbol{J},\qquad \boldsymbol{J}=\begin{bmatrix}\frac{\partial f^1}{\partial u^m} & \cdots & \frac{\partial f^1}{\partial u^m}\\ \vdots & \ddots & \vdots\\ \frac{\partial f^n}{\partial u^1} & \cdots & \frac{\partial f^n}{\partial u^m}\end{bmatrix}$$

度规：

$$g_{ij}=\langle\boldsymbol{r}_i,\boldsymbol{r}_j\rangle=\sum_{k=1}^{n}\frac{\partial f^k}{\partial u^i}\frac{\partial f^k}{\partial u^j},\qquad \boldsymbol{G}=\boldsymbol{J}^{\mathrm{T}}\boldsymbol{J}$$

得到度规后，事实上我们已经可以脱离流形的概念，仅从度规的角度出发去讨论这个空间的性质。

自然标架的每个方向都是V上的向量场。研究它的在临近点的变差

$$\frac{\partial\boldsymbol{r}_i}{\partial u^l}=\sum_{k=1}^{m}\Gamma_{il}^{k}\boldsymbol{r}_k$$

其中$\boldsymbol{\Gamma}\in\mathbb{R}^{m\times m\times m}$，是克氏(Christoffel)记号，下面略去推导过程直接给出

$$\Gamma_{il}^{k}=\frac{1}{2}g^{kj}\left(\frac{\partial g_{ij}}{\partial u^l}+\frac{\partial g_{jl}}{\partial u^i}-\frac{\partial g_{il}}{\partial u^j}\right)$$

其中(g^{kj})是(g_{kj})的逆矩阵。

向量如何平移?

下面导出平移向量场的方程和测地线方程。对摄像头、物体平移时需要将向量平移。测地线方程用于计算光的传播路线。

设光滑向量场$\boldsymbol{v}$定义在区域V上，用自然坐标架表示为

$$\boldsymbol{v}=\sum_{i=1}^{m}v^i\boldsymbol{r}_i$$

$$\frac{\partial\boldsymbol{v}}{\partial u^l}=\sum_{i=1}^{m}\left(\frac{\partial v^i}{\partial u^l}\boldsymbol{r}_i+v^i\frac{\partial\boldsymbol{r}_i}{\partial u^l}\right)=\sum_{i=1}^{m}\left(\frac{\partial v^i}{\partial u^l}+\sum_{k=1}^{n}v^k\Gamma_{kl}^{i}\right)\boldsymbol{r}_i$$

令$\frac{\partial\boldsymbol{v}}{\partial u_l}=0$，可以解得任何平移不变向量场的各分量的梯度。

那么对于向量场沿曲纹坐标表示的曲线$\boldsymbol{u}(t)$， $\boldsymbol{u}'(t)=\sum_{i=1}^{m}\frac{\partial u^i}{\partial t}\boldsymbol{r}_i$方向的平移$\Delta t$，$\boldsymbol{u}$的变化量$\left[\frac{\partial u^1}{\partial t},\ldots,\frac{\partial u^m}{\partial t}\right]\Delta t=[\Delta u^1,\ldots,\Delta u^m]$，$\boldsymbol{v}$的变化量$\Delta\boldsymbol{v}$，其各分量的变化量$\Delta v_1,\ldots\Delta v_m$

$$\langle\Delta\boldsymbol{v},\boldsymbol{r}_\mu\rangle=0,\qquad \text{for } \mu=1,\ldots,m$$

$$\langle\sum_{l=1}^{m}\frac{\partial\boldsymbol{v}}{\partial u^l}\frac{\partial u^l}{\partial t}\Delta t\ ,\boldsymbol{r}_\mu\rangle=0$$

$$\sum_{l=1}^{m}\sum_{i=1}^{m}\left(\frac{\partial v^i}{\partial u^l}\frac{\partial u^l}{\partial t}\Delta t+\sum_{k=1}^{n}v^k\Gamma_{kl}^{i}\,\Delta u^l\right)\langle \boldsymbol{r}_i,\boldsymbol{r}_\mu\rangle=0$$

交换求和次序

$$\sum_{i=1}^{m}\left(\sum_{l=1}^{m}\frac{\partial v^i}{\partial u^l}\frac{\partial u^l}{\partial t}\Delta t+\sum_{l=1}^{m}\sum_{k=1}^{n}v^k\Gamma_{kl}^{i}\,\Delta u^l\right)\langle \boldsymbol{r}_i,\boldsymbol{r}_\mu\rangle=0$$

$$\sum_{i=1}^{m}\left(\Delta v^i+\sum_{l=1}^{m}\sum_{k=1}^{n}\Gamma_{kl}^{i}\,v^k\Delta u^l\right)g_{i\mu}\ =0$$

所以方程为

$$\Delta v^i+\sum_{l=1}^{m}\sum_{k=1}^{n}\Gamma_{kl}^{i}\,v^k\Delta u^l=0$$

对于光线，取$\boldsymbol{u}'=\boldsymbol{v}=\boldsymbol{x}'$，即：沿曲线平移的方向就是向量场方向，即测地线方程。
如果采用迭代解法，不能保证$\|\boldsymbol{x}'\|$不变，如果迭代次数较多时，迭代步长$\|\boldsymbol{x}'\|\Delta t$会变大，失去精度。因此每步迭代应将$x'$正则化（缩放到长度为 1）。也可以对$\boldsymbol{J}\boldsymbol{x}'$正则化。

如何把欧式空间中的物体迁移到非欧空间中的参数坐标体系下

如何把现实世界（三维欧式空间）的物体映射到三维流形的某个**局部**上?

有点类似于我们做过的参数化，那个是在 2 维流形上做的（2 维流形拍扁到 2 维欧式空间）。我们这个相当于过程反过来，3 维欧式空间到 3 维流形。这样看来似乎可以用 ARAP 的方法来做，但是这太困难了。我们就考虑一个很小的物体放到很大的流形中，它的局部是欧式空间，那么我们只要求一个线性变换，克服曲纹坐标的自然标架场不正交的问题就可以了。

利用流形局部是欧式空间的特点
目标：设一个欧式空间中的物体中心为$\boldsymbol{0}$，各个顶点相对中心坐标$\Delta\boldsymbol{x}_i$。将这个物体映射到流形上以$\boldsymbol{u_0}$为中心一系列点$\Delta\boldsymbol{x}_i\to\Delta\boldsymbol{u}_i$

自然切坐标架$\{\boldsymbol{r}_1,\dots,\boldsymbol{r}_m\}$，由已知的度量矩阵 $\boldsymbol{G}$ （取中心u_0处的$\boldsymbol{G}$）施密特正交化得到一组正交基$\{\boldsymbol{t}_1,\dots,\boldsymbol{t}_m\}$，显然它是$\{\boldsymbol{r}_1,\dots,\boldsymbol{r}_m\}$的线性组合，设$\{\boldsymbol{t}_1,\dots,\boldsymbol{t}_m\}=\{\boldsymbol{r}_1,\dots,\boldsymbol{r}_m\}\boldsymbol{S}$，若一个向量$\boldsymbol{p}_i$，它在正交基下$\{\boldsymbol{t}_1,\dots,\boldsymbol{t}_m\}$的坐标为$\Delta\boldsymbol{x}_i$（也就是读入的 obj 文件中的顶点坐标），在自然坐标架$\{\boldsymbol{r}_1,\dots,\boldsymbol{r}_m\}$下的坐标为$\Delta\boldsymbol{u}_i$，则

$$\boldsymbol{p}_i=\{\boldsymbol{t}_1,\dots,\boldsymbol{t}_m\}\Delta\boldsymbol{x}_i=\{\boldsymbol{r}_1,\dots,\boldsymbol{r}_m\}\boldsymbol{S}\Delta\boldsymbol{x}_i=\{\boldsymbol{r}_1,\dots,\boldsymbol{r}_m\}\Delta\boldsymbol{u}_i$$

所以

$$\Delta\boldsymbol{u}_i=\boldsymbol{S}\Delta\boldsymbol{x}_i$$

如果还对物体作了旋转变换$\boldsymbol{R}\in\mathbb{R}^{m\times m}$，则

$$\Delta\boldsymbol{u}_i=\boldsymbol{S}\boldsymbol{R}\Delta\boldsymbol{x}_i$$

在同一点，由$\boldsymbol{G}$按照固定顺序做施密特正交化得到的正交基是唯一的。所以个映射的结果也是唯一的。如果不作旋转变换，默认的正交坐标架就是施密特正交化得到的正交基
求得S

$$\boldsymbol{\beta}_i=\boldsymbol{r}_i-\sum_{j=1}^{i-1}\langle r_i,t_j\rangle\boldsymbol{t}_j=\boldsymbol{r}_i-\sum_{j=1}^{i-1}\sum_{k=1}^{m}S_{kj}g_{ik}\boldsymbol{t}_j=\boldsymbol{r}_i-\sum_{j=1}^{i-1}\sum_{k=1}^{m}S_{kj}g_{ik}\sum_{l=1}^{m}S_{lj}\boldsymbol{r}_l=\boldsymbol{r}_i-\sum_{l=1}^{m}\sum_{j=1}^{i-1}\sum_{k=1}^{m}S_{kj}g_{ik}S_{lj}\boldsymbol{r}_l$$

$$t_i = \frac{\beta_i}{\sqrt{\langle \beta_i, \beta_i \rangle}}$$

如何局部渲染

给了摄像机的三个向量$Camera.\boldsymbol{x}, Camera.\boldsymbol{y}, Camera.\boldsymbol{z}$，摄像机中心位置$Camera.\boldsymbol{pos}$，如何知道一个参数坐标为$\boldsymbol{u} = (\boldsymbol{u}^1, \boldsymbol{u}^2, \boldsymbol{u}^3)$在屏幕上**相对屏幕中心**的位置$(p_x, p_y)$?

$$p_x = \frac{dot(\boldsymbol{u} - Camera.\boldsymbol{pos}, Camera.\boldsymbol{x})}{dot(\boldsymbol{u} - Camera.\boldsymbol{pos}, Camera.\boldsymbol{z})}$$

$$p_y = \frac{dot(\boldsymbol{u} - Camera.\boldsymbol{pos}, Camera.\boldsymbol{y})}{dot(\boldsymbol{u} - Camera.\boldsymbol{pos}, Camera.\boldsymbol{z})}$$

在欧式空间中的摄像机变换就是如此（与矩阵表示等价）

其中 dot 就是点积运算，不同的是由于这里用的是参数坐标下表示的向量，所以点积运算有所不同

$$dot(\boldsymbol{a}, \boldsymbol{b}) = \boldsymbol{a}^{\mathrm{T}} \boldsymbol{G} \boldsymbol{b}$$

由于尺度问题，还要对p_x, p_y做适当比例缩放，才能看到合适大小的图像

不过，这里还可以再整理一下，依然是矩阵变换：

把 Camera.x, Camera.y, Camera.z 按行排成一个矩阵，$\boldsymbol{T}$。

则

$$\boldsymbol{p} = \boldsymbol{TG}(\boldsymbol{u} - Camera.\boldsymbol{pos})$$

假设摄像机是由初始给定的$(1,0,0), (0,1,0), (0,0,1)$向量，经过旋转$\boldsymbol{R}$并经过正交化变换$\boldsymbol{S}$得到的

$$\boldsymbol{T} = (\boldsymbol{SR})^{\mathrm{T}}$$

我们就只需要S和R就能表示正交的摄像机坐标系了，而不需要摄像机向量。这是一个备用的摄像机的表示方案：一方面，可能对于 OpenGL 来说，计算矩阵乘法会更快，我们写的也更简便；另一方面，摄像机平移时，为了避免向量平移方法的误差导致摄像机向量逐渐不正交，还需要每迭代几次就进行一些额外的操作以保证摄像机向量正交，如果我们能直接计算出旋转矩阵的平移，就不存在这个问题，非常方便。（这要看旋转矩阵的计算是否容易实现，如果很难实现我们就不这样做）

光线追踪渲染

~~至此，这些结果已经能用来渲染非欧空间中看到的物体的几何关系。测地线方程仅仅用来判断“能看到什么”。而对于光源、亮度，依然不能渲染-现有的光线追踪不能够得到光照的信息。*这是因为非欧空间中点光源光强不满足平方反比的规律！所以一般的渲染方程不能用！*~~

~~我们计算光线轨道时，还需要同时计算一个类似散度的东西。TODO…~~

更正一下上面的观点。渲染方程依然是能用的，用采样积分的方法仍然可以直接算。（因为蒙特卡罗积分是基于采样的，采样的光线的发散，采样到光源的概率减小，可以体现光强的减弱）。不过几何项确实不能用，也就是说，对于面光源，不能利用快速地计算，必须挨个采样。这意味着采样密度要比一般的光线追踪还要高。（当然，因为光线不沿“直线”传播，我们本来就没法快速计算面光源的贡献）