

每周工作总结

本周主要进行了以下工作：

通过网上的资料学习 AR 相关算法的原理。

学习 Unity 的使用核操作。

具体内容：

一、AR 相关

AR 识别追踪，根据所需识别的图像内容，大致分为两个技术流派：**Marker based AR** 以及 **MarkerLess AR**。**Marker** 识别是使用一个较简单的标识，通常是正方形并且四边都清晰可辨。这类识别算法相对简单，通常用模板匹配算法解决，效果也不错，此方面经典实现可以参考早期 **ARToolKit** 引擎的 **Marker** 识别流程。

随着进一步发展，用更复杂的图像来代替 **Marker**，简单的识别技术已经不适应现有需求。于是就出现了更复杂更通用的 **MarkerLess AR** 的识别。**MarkerLess AR** 的识别追踪主要分为特征点检测、图片识别和追踪（姿态估计）三个步骤。图片识别的过程又分为特征点匹配和图片相似度度量两步。

追踪的过程就是估计被摄物体 3D 姿态及 3D 引擎渲染对应虚拟物体姿态的过程。每一次的追踪，都依赖于前述的特征点提取、匹配、及计算单应性变换。得到单应性变换，就等于得到了被摄物体 3D 现实坐标点集与二维屏幕成像坐标点集。通过得到的匹配点集以及标定获得的摄像机内参数，使用求解透视 N 点问题的算法，就可以求出摄像机外参数，即摄像机实时姿态，随即便可估算出被摄物体 3D 姿态。最后一步，使用 3D 引擎渲染被摄物体上的 3D 虚拟物体，并且把虚拟物体的姿态根据上面估计出的实际物体的姿态矩阵进行变换。

现实世界中所有对象具有三维属性，可以将事物抽象为一个物点，具有空间属性坐标，将三维空间目标与二维平面图形连接的纽带就是坐标，在经历投影变换等降维操作后，虚拟信息由初始的三维坐标形式最终转换为用户屏幕上二维坐标形式。综合来说分为以下几个步骤：

- 1) 数据预处理
- 2) 内参数标定
- 3) 位姿信息求解
- 4) 虚拟信息叠加

要描述了在用户视角下，用户视野内一个物点（事物抽象）Q 最终在投影平面上的投影关系，上述模型主要涉及不同坐标系转换关系的确定。

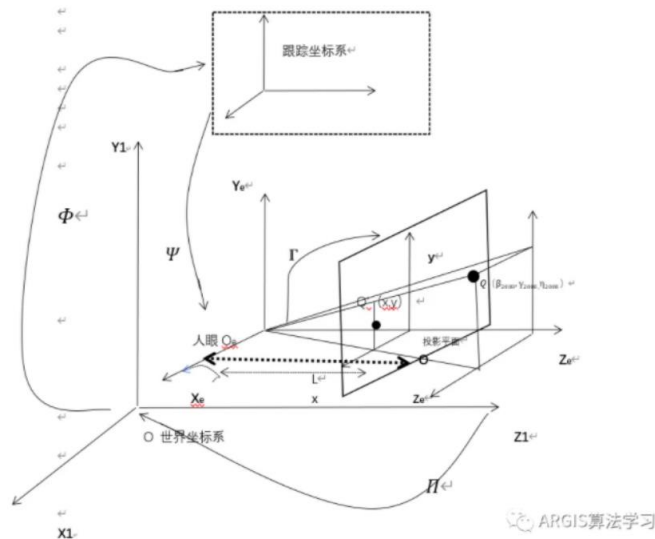
1、场景三维特征点坐标系到世界坐标系转换关系 Π 用于将场景三维特征点的经纬度坐标转换为世界坐标系的描述。

2、世界坐标系到动态追踪坐标系的转换关系 Φ 用于确定场景三维特征点相对于世界坐标系的位置和方向。

3、动态追踪坐标系到人眼坐标系的转换关系 Ψ 确定人眼位置相对真实室内场景中的位置和姿态。

4、人眼坐标系到投影平面坐标系的转换关系 Γ 确定将场景三维特征点坐标转化为二维图像透视投影和姿态。

大致原理如下图所示：



二、Unity 相关

学习了 Unity 中的阴影系统和粒子系统。

Unity 的光源可将游戏对象的阴影投射到其自身的其他部分或其他附近的游戏对象上。阴影为场景添加了一定程度的深度和真实感，因为它们可以显示游戏对象的比例和位置，否则游戏对象看起来显得扁平。

给定光源的阴影是在最终场景渲染期间确定的。当场景渲染到主摄像机视图时，视图中的每个像素位置都将转换为光源的坐标系。然后，像素与光源的距离将与阴影贴图中的对应像素进行比较。如果像素比阴影贴图像素更远，那么它很可能被另一个游戏对象遮挡住光线，因此不会获得光照。为了防止阴影暗斑，可在阴影贴图图中的距离上添加 **Bias** 值，从而确保边线上的像素确实符合比较预期，或者确保在渲染到阴影贴图时游戏对象可以沿着自身法线插入一点点。启用阴影后，这些值由 **LightIn spectator** 窗口中的 **Bias** 和 **Normal Bias** 属性设定。不要将 **Bias** 值设置得太高，因为游戏对象附近的阴影周围区域有时会被错误地照亮。这会导致阴影脱离游戏对象，使游戏对象看起来好像飘在地面上方。

粒子是由粒子系统大量呈现或移动的简单小型图像或网格。每个粒子代表一小部分流体或无定形实体，而众多粒子将共同营造出完整的实物感。以烟幕云团为例，每个粒子都具备细小烟雾的质感，自身形态类似于微型云团。在场景内，大量的这些微型云团通过排列组合，营造出体积更大的整体云团效果。发射和生命周期设置会影响系统的整体行为，不过粒子个体也会随时间变化。每个粒子都有其速度矢量，该矢量决定了粒子在每一帧更新期间移动的方向和距离。系统自身采用的作用力和重力或者地形上将粒子吹散的风区都可以改变该速度。每个粒子的颜色、大小和旋转方式也可能在其生命周期内变化，或与其当前的移动速度成比例。粒子的颜色包括 **Alpha**（透明度）分量，因此粒子可逐渐淡入和淡出，而非突然出现和消失。多种粒子动态性混合运用时，可以将多种流体效果模拟得栩栩如生。例如，利用稀薄的发射形状，使水粒子单纯受重力下落并逐渐加速，即可模拟瀑布。火堆冒出的烟雾往往会上升、扩散并最终消逝，所以系统应为烟雾粒子设置升力，并随时间的推移增大其体积和透明度。